Planning and scheduling in process industries considering industry-specific characteristics

Kilic, Onur Alper

Planning and scheduling in process industries considering industry-specific characteristics

Onur Alper Kılıç

Publisher:	University of Groningen Groningen, The Netherlands
Printer:	Ipskamp Drukkers B. V. Enschede, The Netherlands
ISBN:	978-90-367-5116-2 (book) 978-90-367-5117-9 (e-book)

This document was prepared using LaTeX.

Planning and scheduling in process industries considering industry-specific characteristics

Proefschrift

ter verkrijging van het doctoraat in de
Economie en Bedrijfskunde
aan de Rijksuniversiteit Groningen
op gezag van de
Rector Magnificus, dr. E.Sterken,
in het openbaar te verdedigen op
maandag 10 oktober 2011
om 11.00 uur

door

Onur Alper Kılıç

geboren op 29 januari 1981
te Düzce, Turkije

Promotores:	Prof. dr. D. P. van Donk Prof. dr. J. Wijngaard
Beoordelingscommissie:	Prof. dr. R. H. Teunter Prof. dr. ir. J. C. Fransoo Prof. dr. R. Güllü

To my wife Eda.

Acknowledgements

It was an exhilarating process to take this body of work from an idea to a finished book. It was, however, not easy, and only made possible by the support of those around me. I would like to take this opportunity to thank them.

I am forever indebted to my supervisors Dirk Pieter van Donk and Jacob Wijngaard. It was an excellent opportunity to work with them. They kindly granted me the freedom to explore on my own, and at the same time provided the support and guidance when I needed. I would like to express my sincere gratitude to Dirk Pieter van Donk. His enthusiasm towards research was contagious and motivational. His guidance and trust always kept me on track and gave me the confidence to overcome many obstacles in the course of this research. It was his understanding and patience which made the completion of this thesis possible. I am deeply grateful to Jacob Wijngaard for being a fabulous supervisor. His sharp insights and perception of the things that matter improved my research in many ways. He will always be my perfect role model as a person and as a scholar, and it is my great honor to be mentored by him.

I would like to thank Armağan Tarım. It was his enthusiasm which inspired me to pursue graduate studies in the first place. He always encouraged and supported my work through my doctoral pursuit, and also actively contributed to this thesis.

I am grateful to Martin Grunow and Renzo Akkerman for their warm welcome and hospitality during the course of my visit to the Technical University of Denmark. I appreciate their lively discussions and insightful comments. Their collaboration was an important contribution to this body of work.

I would like to offer my gratitude to the members of my doctoral committee: Ruud Teunter, Jan Fransoo, and Refik Güllü. They kindly accepted to read my thesis, and provided valuable and constructive comments which brought significant improvements to its final version.

I would also like to thank all my warm-hearted and cool-headed friends which are too many to mention here. They kept me engaged and entertained along the way, and made Groningen a very special place over all those years. I would like to give special thanks to my dear friends and paranymphs: Remco Germs and Kristian Peters. They were of great help in many ways including writing the Dutch summary of this thesis.

Finally, I must thank my most devoted and avid supporters: my parents and my wife. I always felt their support when I needed despite the geographical distance. There is no way that I can express my gratitude to Eda – the compass of my life. This thesis is dedicated to her.

Onur Alper Kılıç
Ankara, August 2011

onuralperkilic@gmail.com

Contents

Chapter 1

Introduction

Abstract. *This chapter is devoted to the main concepts underlying the work carried out in this thesis. An overview of process manufacturing is provided, which is followed by a discussion of current trends & challenges in process industries which motivated the development of this thesis.*

1.1 An overview of the process industry

The American Production and Inventory Control Society (APICS) defines process manufacturing as “production that adds value to materials by mixing, separating, forming and/or chemical reactions” (Cox et al, 1995). The definition indicates that process manufacturing is mainly characterized by the type of operations that take place within the manufacturing process. These operations are usually applied on non-discrete materials, and, technologically, they necessitate installations which require large capital investments.

Process manufacturing is common in many industries involving food, chemical, pharmaceutical, and consumer packaged goods. These industries are amongst the largest manufacturing sectors worldwide. For instance, the food-processing industry is the largest manufacturing sector in the European Union in terms of turnover and employment (CIAA, 2010).

The nature of the materials and the operations involved in process manufacturing are quite different as compared to those in discrete manufacturing. This difference clearly shows itself in the terminology used. For example, in process manufacturing, ingredients replace parts, recipes replace bill of materials, and batches replace units as opposed to discrete manufacturing. The difference in the terminology is inherited from the manufacturing practice since the aforementioned terms characterize distinct manufacturing approaches.

There are two basic types of process manufacturing: flow and batch. Flow processing refers to "a lotless production in which products flow continuously rather than being divided". Flow processes are common in systems where a limited number of products are produced in high volumes following rather standardized processing steps. Batch processing is defined as "a manufacturing technique in which parts are accumulated and processed together in a lot". Batch processes are common in systems offering a variety of products which usually undergo distinct processing steps (Cox et al, 1995). There are also processing systems which involve flow and batch type sub-processes. These are often referred to as semi-continuous (or semi-batch) processes (Kallrath, 2002).

The process industry has long been acknowledged as an industry where only a few products are produced following standardized flow type production operations. Following recent trends, however, the process industry has experienced growing logistical demands, growing variety in products, and more intense competition. Thus, the traditional positioning of the processing industry has significantly evolved towards more market oriented strategies and make-to-order policies (Dennis and Meredith, 2000; Van Donk, 2001).

The change in the marketing environment has also affected the manufacturing environments by necessitating the use of multi-product and multi-purpose processing systems offering high flexibility. Such systems have provided a means to meet the increasing demands with regard to the larger number of different products. They have, however, intensely affected planning and scheduling of operations because they require the coordination of a limited set of equipment and resources to undertake a variety of tasks. This has resulted in the need for effective scheduling methods tailor made for process industries.

This thesis is motivated by the aforementioned observation and concentrates on planning and scheduling in the process industry. In the following sections, we first discuss some important characteristics of the process industry and provide a perspective of the main problems addressed in the context of the thesis. Then, we review and discuss the literature on scheduling approaches in process industries. Finally, we provide the research objectives and present an outline of the thesis.

1.2 Characteristics of the process industry

There is a large amount of research centered on the characteristics of process industries. In particular, the differences between process manufacturing and discrete manufacturing in terms of demand management, production process, quality, and production planning and control have been elaborated by many authors (see e.g. Taylor et al, 1981; Fransoo and Rutten, 1994; Ashayeri et al, 1996). Nevertheless, not all of those characteristics have immediate effects on planning and scheduling processing systems. In what follows, we rather concentrate on some characteristics of the process industry that often lead to difficulties in planning and scheduling processing systems, and need to be treated in a peculiar manner. We refer to Fransoo and Rutten (1994) and the references therein for a detailed summary and discussion of the general characteristics of the process industry.

Raw materials. In the process industry, raw materials have a prominent effect on the planning and scheduling activities. This is mainly because the process industry obtains raw materials from mining and agricultural industries. The quality, and thus, the yield of these raw materials are often subject to variability (Rice and Norback, 1987; Gunasekaran, 1998). Therefore, production plans must account for the variability of raw materials in terms of quality and availability (Taylor et al, 1981).

Recipes. The materials concerned in the processing industry are characterized by their recipes. These recipes define the ingredient compositions of materials, and also specify the succession of production steps that materials undergo. There could often be several potential recipes for intermediates and end products since many products are produced from a few raw materials in process industries. Thus, the decisions on which recipes to use are usually made in connection with seasonal considerations, the scarcity of raw materials, and the availability of processing and storage facilities (Cokins, 1988). It is important to note that these decisions have immediate effects on scheduling production operations. That is because not only the ingredient compositions of materials but also the course of processing operations may vary according to product recipes.

Perishability. The process industry use raw materials, intermediates, and also end products which are often subject to perishability. This necessitates the careful handling of inventories throughout production processes and shipping, and leads to specific constraints on planning and scheduling. In particular, perishability has an important effect on storage operations. Storage operations lead to better utilization of resources by decoupling consecutive processing stages. Hence, limitations on the storage operations constrain the extent to which the upstream and downstream production units can be decoupled. In presence of storage time constraints, inappropriate scheduling decisions may result in a high number of setups, blocking or starvation throughout the stages, waste of intermediates, and thus, degrade the overall system performance significantly. Furthermore, perishability dictates the segregation of batches in order to avoid degradation and makes production scheduling more difficult (Akkerman and Van Donk, 2009).

Traceability. It is often required to have tracking systems in the process industry to trace back the origin of materials through the processing system. Traceability is an important aspect especially in the food processing industry where it is conducive to food safety. Food safety has become an important concern in recent years due to food scandals and incidents. The main benefit of traceability lies in handling product recalls in an effective way when a problem has been identified (Rong and Grunow, 2010). Traceability necessitates a perpetual control of material flows through production and storage operations. Thus, maintaining traceability becomes more challenging as material batches are merged and/or separated in successive processing and storage operations.

Storage. It is obvious that storage limitations are common in all types of manufacturing systems. However, they are considered to be more critical in process industries. This is mainly due to the fact that the nature of storage is rather different as compared to other industries. In most production environments, storage space is mainly a buffer or a warehouse where items are stocked all together. In the process industry, however, storage operations are carried out by a number of discrete storage units (i.e. tanks, vessels, silos) which can be used only for a single type of material at a time. This results in capacity constraints for each individual storage unit rather than an aggregate storage capacity constraint. Storage units are either dedicated to certain material, or flexible and can be used for several materials. The latter requires the assignment of storage units to materials, as is the case for processing units. The aforementioned characteristics of storage units also interfere with perishability and traceability issues. These, all together, translate into complex restrictions on both processing and storage operations.

Setups. Production setups are often required to prepare the machinery for new production runs. As compared to other industries, setups take significant amounts of time and effort in the process industry. Moreover, besides the setups for configuring the settings of production units, there are setups for the cleaning requirements. It is also common that setups are sequence-dependent due to the differences in the product and process specifications among different materials (Van Wezel et al, 2006). For instance, in dairy industries it may be more favorable to process products sequenced from low to high fat concentrations, because there is a little or no effort needed to clean processing units following this sequence. Especially when the capacity utilization is high, setups may significantly affect the system performance. The main issue here is the trade-off between the setup time and the lead time. That is, significant setups stimulate larger production runs, but in return they delay consequent processing runs (Potts and Kovalyov, 2000).

The characteristics of the process industry we mentioned in this section have significant effects for their own sakes. However, it is important to note that one has to deal with the combination of these characteristics when scheduling processing systems. In the next section, we provide a review of the literature on scheduling approaches in the context of the process industry.

1.3 Scheduling in the process industry

The efficiency of production is at the forefront of the current competitive global environment. In this context, scheduling plays a major role by coordinating and integrating production facilities and resources. The scheduling of processing systems mainly involves the decisions regarding the allocation of production facilities and resources to all types of operations, and the timing of those operations. In this section, we briefly discuss the scheduling approaches used in the process industry. As mentioned earlier, flow and batch processes are characterized by different degrees of material diversity and routing complexity. Because of this reason, the scheduling practices utilized in those processes are quite different from each other. Here, we review the scheduling literature directed towards flow and batch processes individually.

1.3.1 Flow processes

Flow process industries usually involve the manufacturing of a small number of products with a limited variety. The processing steps are rather standard, products mainly follow the same routing, and the flexibility of processing units is quite limited. The customer demands are relatively stable (Cooke and Rohleder, 2006).

The limited product and routing variety in flow processes often postulate a bottleneck production operation which dominates others in terms of capacity utilization. Furthermore, the lack of flexibility abates the need of considering multiple machines. Therefore, it is justifiable to concentrate only on this bottleneck stage with a single processing unit when scheduling flow processes (Fransoo, 1993). Consequently, scheduling approaches in flow processes mainly follow single-stage, single-machine, multi-product, lot-sizing and scheduling problems which are also commonly used in discrete manufacturing. We can acknowledge two main research lines in this domain which takes different approaches in expressing time and customer demands.

The first line of research is centered on the economic lot sizing problem (ELSP) (Rogers, 1958). The ELSP is mainly the multi-product capacitated version of the well-known economic order quantity (EOQ) problem. The assumptions of ELSP are as follows: demand and production is deterministic and continuous, setup time and cost are sequence independent, and backlogging is not allowed. The ELSP aims to find a production schedule that minimizes the sum of holding and setup costs per unit time. An ELSP schedule refers to the length and position of the production cycles associated with the set of products to be produced. The ELSP is known to be a difficult problem. In fact, even checking the feasibility of a given ELSP schedule is NP-hard (Hsu, 1983). Thus, a plethora of literature has been emerged to tackle this problem. These research efforts resulted in many efficient approaches to solve the problem heuristically. Among those, we can mention the common cycle, the basic period, the extended basic period, and the multiple cycle approaches. The interested reader is referred to Elmaghraby (1978) and Raza and Akgunduz (2008) for an overview of the problem and associated solution approaches.

The second line of research is concentrated on finite-horizon, periodic, and time-varying counterparts of the ELSP. A variety of problems has been considered in this area. The fundamental assumptions of these models are as follows: planning horizon is finite and comprised of a number of time periods with deterministic lengths, demand over periods is deterministic and time-varying, production rates are fixed, setup costs are sequence independent, setup times are negligible, and backlogging is not allowed. The objective is to find a production schedule that minimizes the sum of holding and setup costs within the planning horizon. Here, a schedule specifies the products and lot sizes associated with each time period within the planning horizon. The most well-known problems in this domain are: the capacitated multi-item lot sizing problem (CLSP) (Bitran and Yanasse, 1982), the discrete lot sizing problem (DLSP) (Lasdon and Terjung, 1971; Fleischmann, 1990), the continuous setup lot sizing problem (CSLP) (Karmarkar et al, 1987), the proportional lot sizing and scheduling problem (PLSP) (Drexl and Haase, 1995), and the general lot sizing and scheduling problem (GLSP) (Fleischmann and Meyr, 1997). The CLSP is a so-called large-bucket model which is characterized by time periods large enough to accommodate several different products. Practically, periods in large-bucket problems correspond to time slots of weeks (Drexl and Kimms, 1997). The CLSP is often regarded as a medium-term planning problem since scheduling decisions within single time periods are not integrated into the problem. The DLSP is a so-called small bucket problem which is characterized by small time periods in each of which at most one type of product can be produced. The length of the periods in small-bucket problems correspond to hours/shifts (Drexl and Kimms, 1997). The main assumption of DLSP is that in each time period the facility either processes a single type of product at full capacity, or remains idle (i.e. all-or-nothing). Thus it is not necessary to determine lot sizes explicitly. The CSLP relaxes the all-or-nothing assumption and allows continuous lot sizes. The PLSP relaxes the one product per period limitation and allows up to two products per period. The GLSP is an integrated large- and small-bucket problem. The planning horizon is divided into a number of macro-periods each of which is composed of a number of micro-periods. Then, each macro-period is treated as a small-bucket problem, whereas the overall problem is treated as a large-bucket problem. There is an extensive literature on the aforementioned problems and their extensions. Recent overviews of this line of research can be found in Wolsey (2002) and Jans and Degraeve (2008).

Despite the vast majority of the literature focuses on single-stage single-machine multi-product lot-sizing and scheduling problems, some research effort has also been taken towards multi-stage and/or multi-machine extensions of the problems mentioned above. Some variants of the ELSP include parallel-machine ELSP (see e.g. Carreno, 1990; Bollapragada and Rao, 1999; Pesenti and Ukovich, 2003), multi-stage ELSP (see e.g. El-Najdawi and Kleindorfer, 1993; Dobson and Yano, 1994; El-Najdawi, 1997; Ouenniche and Boctor, 1998; Ouenniche et al, 1999; Ouenniche and Bertrand, 2001), and multi-stage and multi-machine ELSP (see e.g. Torabi et al, 2005; Jenabi et al, 2007). Some variants of the CLSP and the DLSP include parallel-machine CLSP (see e.g. Ozdamar and Birbil, 1998; Ozdamar and Barbarosoglu, 1999; Kang et al, 1999; Clark and Clark, 2000; Belvaux and Wolsey, 2000), parallel-machine DLSP (see e.g. Salomon et al, 1991; Jans and Degraeve, 2004; De Matta and Guignard, 1994a,b, 1995), and multi-stage CLSP (see e.g. Tempelmeier and Derstroff, 1996; Franca et al, 1997; Katok et al, 1998; Ozdamar and Barbarosoglu, 2000; Berretta and Rodrigues, 2004; Sahling et al, 2009). It is important to note that such problems also involve the consideration of the assignment of production units and the interdependency between processing stages. From this point of view, they can be regarded as steps towards batch processing systems. Nevertheless, they do not account for lower and upper bounds on processing times/quantities which are prominent in batch processes. Furthermore, they mainly concentrate on processing operations and do not integrate other resources such as storage units into the scheduling problem.

1.3.2 Batch processes

Batch process industries commonly produce a large number of products in small quantities following customer orders. The number of processing operations is large and routing complexity is high (Rippin, 1991). In order to provide flexibility, multi-purpose production units are preferred. The size of each batch is often bounded from below and above due to technological constraints. This may necessitate performing successive batch operations for the same material – often referred to as a product campaign (Kallrath, 2002). It might be evident from the aforementioned characteristics of batch processes that it is often not realistic to assume that a single bottleneck operation dominates the rest of the production operations.

The batch process scheduling problem is to find a production schedule optimizing a time-based (e.g. makespan, tardiness) or a financial (e.g. cost, revenue) objective given the configuration of the processing system and the demand requirements. The configuration of the processing system involves the set of available resources, and the product recipes which define the processing operations required to produce each intermediate and end product. The demand requirements involve the size and due dates for each customer order. In this context, the fundamental decisions involve the number and the size of batches associated with each intermediate and end product, the assignment of batches to resources, and the timing of processing operations.

The vast majority of research contributions in the area of scheduling batch processes have been emerged in response to the short-term scheduling needs of large-scale chemical plants. Two types of batch processes are usually distinguished based on their process layout. The first type is the multi-stage processes where one or more processing units work in parallel in each stage. In multi-stage processes, each batch is processed in a succession of stages defined through the associated product recipe. Batches are not mixed or split, and the batch identity is, thus, maintained throughout the whole process. The second type is the network processes where processing operations are connected through an arbitrary network structure. In this type of processes, product recipes are rather complex and batches are mixed and split through the processing operations. There is a large body of literature on the short-term scheduling of multi-stage and general network type batch processes. The research in this domain mainly aims at developing a general purpose framework which can systematically characterize and accommodate all relevant aspects in batch processing systems. We can mention equipment connectivity, storage settings, material transfers, batch size and processing time settings, demand characteristics, setup considerations, labor and utility constraints, maintenance operations, and operational costs as some of those aspects (Mendez et al, 2006). Batch process scheduling problems are predominantly formulated as exact mixed integer linear programming (MILP) models. There is a variety of modeling approaches which vary in terms of their time, event, and material balance representations. These approaches significantly affect the flexibility and the computational performance of the associated optimization models. The interested reader is referred to Kallrath (2002); Floudas and Lin (2004), and Mendez et al (2006) for comprehensive reviews on models and methods employed for batch process scheduling problems.

There is also research efforts concentrated in adopting lot sizing and scheduling approaches to deal with batch processes. These research contributions mainly build on existing single-stage and single-machine lot sizing and scheduling formulations. They use a stepwise setup cost function which is not only dependent on the lot size but also on the number of batches needed to fill the associated lot sizes (see e.g. Lippman, 1969; Lee, 1989; Pochet and Wolsey, 1993). These approaches can correctly account for setup costs in batch processes. However, they are limited to single-stage and single-machine environments, and they do not capture the setup times required when switching from one batch to another.

1.4 A critical overview of the literature

In the previous sections, we summarized some specific characteristics of the process industry which may have prominent effects on scheduling, and outlined some research domains which are relevant in the context of scheduling in the process industry. In this section, we provide a critical overview of the outlined literature while taking into account the given specific characteristics of the process industry. We mainly analyze the extent to which those characteristics are captured in the literature.

The process industry is characterized by raw materials which are often subject to seasonal supply and yield variability. These characteristics of raw materials may significantly influence production planning and control. Crama et al (2001) pointed out that if the key raw material is a scarce resource, then demand satisfaction may not be enforced anymore. Schuster and Allen (1998) considered this problem and showed that, in such cases, management of raw material becomes an essential task, and it is necessary to employ an admission policy in order to maximize profit.

The flexibility of recipes in the process industry necessitates the interaction of recipe selection and scheduling decisions. The recipe selection problem itself falls into the category of the well-known blending problems. The blending problem aims at finding a minimum cost mix satisfying a set of quality related attributes (Crama et al, 2001). Despite the extensive literature on standalone blending problems, there are only a few examples where the blending problem is incorporated in operational planning and scheduling. There has been some work on multi-period production planning problems where the purchasing and production quantities are periodically determined while taking blending operations into account (see e.g. Williams and Redwood, 1974; Rutten, 1993). However, they only concentrate on material balances, and do not consider the economies of scale in batch production.

A more relevant line of research addresses grade selection and blending problems where a set of basic grades are selected to be used in processing a large mix of products (see e.g. Karmarkar and Rajaram, 2001; Akkerman et al, 2010). However, these studies assume unlimited production and storage capacities. The pooling problem is also a classic example of the blending problems. It is particularly important in the petrochemical industry (see e.g. Baker and Lasdon, 1985; Amos et al, 1997; Audet et al, 2004). The pooling problem refers to a situation where materials with different quality characteristics are merged in a series of pools (e.g. tank or vessel) such that the quality specifications of the blends satisfy a given set of quality requirements. Nevertheless, the pooling problem is predominantly regarded as a design problem and it is not integrated with the scheduling decisions.

Perishability is a major concern in the process industry. We can acknowledge two types of perishability which are characterized by fixed and random storage times (Nahmias, 1982). The quality (e.g. texture, color, taste or nutrient content) of materials with fixed storage time does not significantly change during some specified time period. However, materials immediately deteriorate after the end of this period. Thus, they can be kept in stock for some time after which they must be disposed of. When the storage time is random, on the other hand, the quality of materials gradually decreases following to a stochastic process and eventually hits to a minimum acceptable level. Thus, they can be retained in stock during a random storage time after which they should be discarded. Both types of perishability are extensively investigated in the inventory literature under various control policies. An overview of this research can be found in Silver et al (1998). In this context, uncapacitated lot sizing problems have also been extended with the issue of perishable inventories (see e.g. Hsu, 2000; Chu et al, 2005). However, there are only a few examples of lot sizing and scheduling problems involving perishability. These are mainly concentrated on the ELSP and assume fixed storage times (Silver, 1989, 1995; Sarker and Babu, 1993; Goyal, 1994; Viswanathan, 1995; Viswanathan and Goyal, 1997, 2000, 2002; Soman et al, 2004b). These studies propose a variety of heuristics to deal with perishability, and show that this limitation may significantly affect the optimal production schedule.

In the literature on batch process scheduling, storage time constraints are reflected by storage policies which define the maximum amount of time that a batch can wait between its release from one processing operation and its start in the next processing operation (Mendez et al, 2006). These storage policies are unlimited wait (UW), finite wait (FW), and zero wait (ZW). Among those FW is the most general storage policy and it reflects perishability with fixed storage times. These policies are successfully implemented in general purpose batch process scheduling models (see e.g. Kondili et al, 1993a; Schilling and Pantelides, 1996; Mockus and Reklaitis, 1997; Ierapetritou and Floudas, 1998). However, the impact of storage time limitations is more severe when the storage capacity is also limited. This issue is considered in the context of multi-stage batch processes (see e.g. Sundaramoorthy and Maravelias, 2008). The problem is even more demanding when batches of the same material are mixed and split through the production process. To the best of our knowledge, there exists no work addressing the combination of storage capacity and storage time limitations.

There has been an increasing societal concern on food safety issues in response to food safety crises, such as mad cow disease, bird flu, and salmonella in recent years (Rong and Grunow, 2010). This has necessitated the use of traceability systems in order to trace back materials through the chain and handle product recalls effectively (Thakur et al, 2010). The operations management and operations research literature has only recently started to take over this issue. Dupuy et al (2005) proposed a batch dispersion model to optimize traceability in food industry by minimizing the batch size and the extent of batch mixing. Tamayo et al (2009) considered a raw material dispersion optimization model to reduce the number of batch recalls in case of a crisis. Thakur et al (2010) proposed a multi-objective optimization model that provides an effective method for minimizing the food safety risk caused by lot aggregation at a grain elevator. However, none of those models take into account the additional cost/time needed in managing traceability. There are a few papers which reflect the effects of the decisions taken to reduce dispersion on the operational performance. Wang et al (2009) and Wang et al (2010) used the concept of batch dispersion in the context of the economic production quantity (EPQ) model. They developed optimization models which also consider potential logistics efforts due to product recalls Rong and Grunow (2010) use the chain dispersion concept while using an uncapacitated lot sizing problem. Nevertheless, none of these research efforts consider traceability in the context of process scheduling with capacitated resources.

Due of their practical relevance, storage limitations have been widely investigated in the literature. Nevertheless, in lot sizing applications, they are usually modeled as an aggregate warehouse capacity constraint. For example, Anily (1991) and Gallego et al (1996) extended ELSP by limited warehouse capacities. They proposed sophisticated heuristics for the problem, and provided lower bounds on the optimal costs of different inventory policies under storage constraints. Van Vyve and Ortega (2004) considered the case where a fixed cost is charged per number of storage units employed.

In the literature on batch process scheduling, storage capacity constraints are reflected by storage policies (Mendez et al, 2006). These are unlimited intermediate storage (UIS), finite intermediate storage (FIS), and no intermediate storage (NIS). Notice that FIS represents the most general case. These policies are successfully implemented in general purpose batch process scheduling models (see e.g. Kondili et al, 1993a; Schilling and Pantelides, 1996; Mockus and Reklaitis, 1997; Ierapetritou and Floudas, 1998). Nevertheless, it is well-known that the number of storage units, especially in case of non-zero cleaning times, significantly increases the complexity of resulting mathematical models. This effect is even stronger if storage units are flexible, i.e. they can be used for more than one type of material and cleaning operations are needed when switching from one material to another.

Setups are seminal in almost all manufacturing environments. However, they usually require larger time and effort in the process industry. The literature on both lot sizing and scheduling problems and batch process scheduling problems pay a considerable attention to setups. In particular, a significant amount of research effort is devoted to sequence-dependent setups. In case of sequence-dependent setups, when evaluating a given production schedule, one should consider not only the set of products but also the production sequence. Thus, the resulting problem is significantly more difficult than its sequence-independent counterpart. Here we shortly mention some of the work in this research domain.

The research efforts on lot sizing and scheduling problems involving setups can be summarized as follows. Galvin (1987), Lopez and Kingsman (1991), Dobson (1992), Wagner and Davis (2002), and Brander and Forsberg (2005) studied sequence-dependent setups in the context of the ELSP. Dilts and Ramsing (1989), Haase (1996), Kang et al (1999), Laguna (1999), Clark and Clark (2000), Haase and Kimms (2000), Meyr (2000), and Gupta and Magnusson (2005) considered the CLSP with sequence-dependent setups. Cattrysse et al (1993) studied DLSP with setup times. Fleischmann (1994), De Matta and Guignard (1994b), and Salomon et al (1997) considered the DLSP with sequence-dependent setup costs and times. Vanderbeck (1998) formulated the CSLP with fractional start up times. Drexl and Haase (1995), and Drexl and Haase (1996) discussed the PLSP with setup times and multiple machines. Wolsey (1997) studied the CSLP with sequence-dependent setups. Suerie (2006) provided an approach that allows the setup times to be split between two periods. Belvaux and Wolsey (2000), and Belvaux and Wolsey (2001) presented a series of lot sizing and scheduling models, including sequence-dependent costs and times.

The research on batch process scheduling problems has predominantly acknowledged sequence-dependent setups. Most of the research efforts in this domain, even the earlier contributions, considered sequence-dependent setups as one of the basic characteristics of batch processing systems (see e.g. Sahinidis and Grossmann, 1991a; Kondili et al, 1993a,b; Shah et al, 1993; Papageorgiou and Pantelides, 1996a,b). There are also studies which explicitly address modeling sequence-dependent setups (see e.g. Kelly and Zyngier, 2007).

1.5 Real-life implementations

The real-life scheduling problems found in process industries are often large-scale combinatorial optimization problems, and due to the curse of dimensionality, they can hardly ever be solved in reasonable computational times by using exact approaches (Mendez et al, 2006). Therefore, although the research efforts taken in the development of general purpose approaches are valuable from a theoretical point of view, the resulting models cannot readily be used to solve real-life process scheduling problems. This has motivated many researchers to develop alternative solution approaches which are less demanding in terms of computational time in exchange for compromising the optimality. The research efforts in this line often make use of the specific characteristics of the underlying process in order to streamline the optimization models. Here we provide some representative examples of such approaches.

Harjunkoski and Grossmann (2001) studied a real-life problem originating from a steel-making continuous casting plant. In order to solve the problem within a reasonable computational time they employed a three-stage decomposition approach. In each stage of the approach, some parts of the schedule are fixed and used as input in the subsequent stage.

Schwindt and Trautmann (2000) and Neumann et al (2002) propose a rather general decomposition approach. Their approach decomposes the batch scheduling problem into separate batching and scheduling problems. The batching problem determines the number and the size of the batches to be processed. Schwindt and Trautmann (2000) used a heuristic to solve the batching problem which sequentially determines the batch sizes of end products and associated intermediates gradually. Neumann et al (2002) propose a mixed integer non-linear program (MINLP) that minimizes the number of batches weighted by the processing times assuming that all batches of a product have the same batch size. The scheduling problem generates a schedule indicating the timing of processing those batches determined by the batching problem. However, the batch scheduling problem itself is intractable. Consequently, Schwindt and Trautmann (2000) used a branch-and-bound approach, and Neumann et al (2002) employed a relaxation approach to solve the batch scheduling problem heuristically.

Ferris et al (2009) considered the problem of scheduling of multi-stage batch processes. They developed a dynamic decomposition approach that exploits the structure of the underlying problem. Their approach dynamically decomposes the problem into a set of subproblems which are generated by fixing the batch selection, unit allocation, and timing decisions respectively, and make use of a grid computation approach to solve the sub-problems. They showed that the proposed approach can solve problems of realistic size within computational times small enough for practical applications.

1.6 Research objectives

The discussion provided so far in this chapter points out that the process industry has a specific set of product and process characteristics which have a significant impact on the management of production operations. These characteristics are practically important for their own sakes. However, in many production environments, a subset of these characteristics appears together. Thus, planning and scheduling in process industries often concerns the combination of several industry-specific characteristics. It is important to note that these characteristics may closely interact with each other. Let us, for instance, consider storage capacity limitations. As mentioned earlier, these limitations are relatively critical in the process industry due to the use of discrete storage units. However, it is evident that storage limitations become even more critical when the production process involves perishable materials. Here, one needs to allocate the available storage capacity while also considering the age of materials in order to avoid degradation. The same holds for traceability requirements. Traceability prohibits the extent to which materials are merged and separated, and puts a lot of pressure on the utilization of storage units. Therefore, we can conclude that dealing with the combination of such industry-specific characteristics is more challenging than dealing with each of them individually. Furthermore, from a mathematical modeling point of view, it is difficult to translate the combinations of complex operational characteristics into mathematical expressions. Despite the discussion provided here, it appears that while the literature offers a variety of methods which take into account these characteristics individually, the relationship between them has not been fully addressed. This observation was also brought up by Van Donk and Fransoo (2006) and Akkerman (2007). As a result, it is not straightforward to see whether conventional planning and scheduling approaches can be adopted to be used in more complex production environments. Also real-life scheduling problems often suffer from high computational complexity which impedes the practical application of exact modeling approaches. This necessitates the use of computationally tractable heuristic procedures, albeit with possible sacrifice in optimality. Therefore, it requires a careful investigation to determine which modeling approaches and optimization techniques are most appropriate given the nature of the underlying processing systems.

The aforementioned observations motivate this thesis. The aim of the thesis is to add to the knowledge on planning and scheduling in the process industry while concentrating on processing systems where combinations of the specific characteristics elaborated in this chapter are of concern. A strong emphasis is laid on scheduling problems originating from the food processing industry due to the relevance and the criticality of its industry-specific characteristics. It would be fair to state that many characteristics, e.g. raw material availability and yield, flexible product recipes, perishable materials, traceability requirements, storage capacity limitations, and production setups can be considered as ordinary concerns in the food processing industry. It is very difficult, if not impossible, to develop mathematical models which can systematically accommodate all these characteristics simultaneously. Nevertheless, based on the product and the process characteristics of the underlying production environments, often a few of these characteristics become relatively important with respect to others. For instance, let us consider a flour manufacturer that supplies flour products to bakeries and industrial manufacturers. The raw materials used in this manufacturing process are mainly wheat and some other starchy plant foods. The supply of such raw materials is fairly stable and availability is usually not an issue. Furthermore, the yield variability of milling – the main processing operation – is very low. However, due to the large variety of end products limited storage capacity is an important concern. A practical approach used to overcome this issue is to make use of flexible product recipes, and to produce and stock a limited number of intermediate products which are then blended into end products following demand. However, this strategy leads to a production scheduling problem with limited production and storage capacities, and also integrated product design decisions regarding the product recipes of end products. This thesis focuses on practical planning and scheduling problems such as described above, and presents new integrative approaches while making use of both optimal and heuristic procedures. The planning and scheduling problems considered in the thesis are motivated by a variety of practical cases originating from different production environments. These cases have emerged from the long-standing industrial collaborations of the research group where this thesis research was carried out.

The main research objectives of this thesis are: (i) to contribute to the development of mathematical models that can be used as decision aids in scheduling processing systems with industry-specific characteristics, and (ii) to provide some insight into the order acceptance function in the process industry with respect to limitations in raw material availability.

The thesis is organized as a collection of research papers centered around the main research theme. These research papers are devoted to particular problems of practical interest originating from specific production environments. Each individual paper mathematically defines and formalizes a particular problem and develops solution approaches thereof based on well-grounded optimization methods. The majority of the work carried out in this thesis concern deterministic scheduling problems. Nevertheless, the order acceptance function in batch processes is addressed while considering stochastic yield and demand. The research efforts presented in different chapters of the thesis are motivated by similar considerations. Thus, there might be some extent of overlapping in positioning these chapters since they are devised not only as parts of the thesis but also as research papers which can be read individually. In the following section, we provide a brief outline of the contributions of each research paper included in the thesis.

1.7 Thesis outline

The thesis is organized as follows. Chapter 2 addresses the scheduling problem in a two-stage flow process. The essence of the problem lies in the use of flexible product recipes. There is a variety of intermediate products characterized by different product recipes which can be processed into end-products. The problem is to determine a production schedule as well as the set of intermediates to be used minimizing the total operational costs while considering production and storage capacity limitations. The chapter presents a comprehensive MILP model for this problem. The model is applied on the data collected from a real-life case. The results of the numerical study are used to analyze the effects of cost parameters and capacity limitations on the selection of product recipes. Also the trade-offs between capacity limitations and operational costs are investigated.

Chapter 3 considers the detailed short-term scheduling problem in batch processes. The chapter contributes to the literature by extending the conventional discrete time MILP formulation for scheduling batch processes by introducing storage capacity and storage time limitations. The model is applied in a variety of storage configurations involving single/multiple and dedicated/multipurpose storage vessels. By means of a numerical study several examples are illustrated to highlight the impact of storage capacity and storage time limitations on scheduling production and storage operations.

Chapter 4 investigates a process scheduling problem originating from a processing system specialized in evaporated milk products. The processing system has a semi-continuous structure. The layout of the system involves two continuous processing stages connected by a batch-wise standardization step. The production environment has several industry-specific characteristics such as traceability requirements and time- and sequence-dependent cleaning of production units. The chapter presents a two-phase mathematical approach for this problem which successively determines the specifications regarding material flows and builds a complete production schedule. The approach is shown to be efficient by means of a numerical study based on a data set collected from a real-life evaporated milk plant.

Chapter 5 focuses on the issue of raw material availability and yield variability in the context of food processing. A food processing system is considered which processes a single raw material with seasonal supply into several end products. The demand for end products is stochastic. Also, the amount of raw material required to fulfill a customer order is not known with certainty due to the variability in yield. The problem is to decide whether or not to accept an incoming order given the amount of raw material available and the time remaining until the next replenishment epoch of the raw material inventory. The chapter elaborates this decision problem while considering the objective to maximize the expected total revenue. It is shown that the problem can be modeled as a single resource capacity control problem. Because the optimal policy is too complex for practical use, a heuristic approach is proposed based on rather simple decision rules. The performance of the heuristic is then investigated by means of a numerical study, and the effects of yield variability are analyzed.

Finally, Chapter 6 concludes the thesis where the individual chapters are summarized, and possible directions for further research are outlined.

Chapter 2

Capacitated intermediate product selection and blending in the food processing industry

An earlier version of this chapter is published as Kilic, Akkerman, Grunow, and Van Donk (2009), Modeling intermediate product selection under production and storage capacity limitations in food processing, Proceedings of the International Conference on Industrial Engineering and Engineering Management, 1077 - 1081.

Abstract. *This study addresses a capacitated intermediate product selection and blending problem typical for two-stage production systems in the food processing industry. The problem involves the selection of a set of intermediates and end product recipes characterizing how those selected intermediates are blended into end products to minimize the total operational costs under production and storage capacity limitations. A comprehensive mixed integer linear model is developed for the problem. The model is applied on a data set collected from a real-life case. The trade-offs between capacity limitations and operational costs are analyzed, and the effects of different types of cost parameters and capacity limitations on the selection of intermediates and end product recipes are investigated.*

2.1 Introduction

The food processing industry is characterized by divergent product structures where a relatively small number of (agricultural) raw materials are used to produce a large variety of often customer specific end products (see e.g. Akkerman and Van Donk, 2009). Due to the large variety of end products it is often not possible or at least inefficient to produce and stock all end products. A common practice used to mitigate the effect of the product variety on the operational performance in food-processing systems is to produce some or all end products by blending them from a limited number of selected intermediate products (Van Donk, 2001; Soman et al, 2004a; McIntosh et al, 2010). The basic notion of this practice follows the well known principle of postponement which is widely used amongst various industries (Van Hoek, 1999; Venkatesh and Swaminathan, 2003; Caux et al, 2006; Forza et al, 2008). This approach reduces the frequency of processing runs and the storage requirements in the intermediate product level at the expense of additional blending operations in the final production stage. However, this strategy leads to a decision problem involving (i) the selection of a set of intermediates from a large set of potential intermediates usually designed by quality management experts, and (ii) end product recipes which prescribe how those selected intermediates are blended into end products in order to minimize the total operational costs (Rutten, 1993; Akkerman et al, 2010).

The current study seeks to address the aforementioned decision problem. The problem relates to the well-known blending problems, where, given a set of products, the objective is to find a minimum cost mix satisfying a set of quality related attributes. Due to their practical relevance, a considerable amount of work has been done on industry-specific production planning problems involving blending components, such as feedlot optimization problems (see e.g. Glen, 1980; Taube-Netto, 1996), sausage blending problems (see e.g. Steuer, 1984), multi-period production planning problems (see e.g. Williams and Redwood, 1974; Rutten, 1993), and grade selection and blending problems (see e.g. Karmarkar and Rajaram, 2001; Akkerman et al, 2010). However, these studies assume unlimited production and/or storage capacities. The problem we consider in this paper stands apart from the aforementioned literature with regard to two main aspects. First, we capture whether blending of intermediates is required to produce end products by acknowledging the possibility of direct use of intermediates as end products. Secondly, we approach the blending problem by considering the costs and the capacity limitations related to both the production and the storage operations which also affect the selection of the intermediates and end product recipes.

The rest of the paper is organized as follows: In Section 2.2, we provide a detailed description of the production system under consideration. In Section 2.3, we review the related literature. In Section 2.4, we present the mathematical programming formulation of the problem. In Section 2.5, we demonstrate an application of the model for a real-life case. We conduct a numerical study to illustrate the effects of some operational settings on the optimal decisions. Finally, in Section 2.6, we summarize our work and suggest directions for future research.

2.2 Problem description

The production system under consideration involves two production stages: processing and blending. The processing stage involves the production of intermediates. In the blending stage, intermediates are blended into end products following end product recipes which specify the blending proportions of intermediates. The recipe of an end product may involve single or multiple intermediates. In the former case, demands can directly be satisfied from intermediate stocks. In the latter case, however, intermediates are first blended to form end products which are then used to serve demands. Figure 2.1 illustrates a small example of such a system involving two selected intermediates and three end products where circles and rectangles represent materials and production operations respectively. Notice that two of the three end products in the example require blending operations, whereas the last one does not.

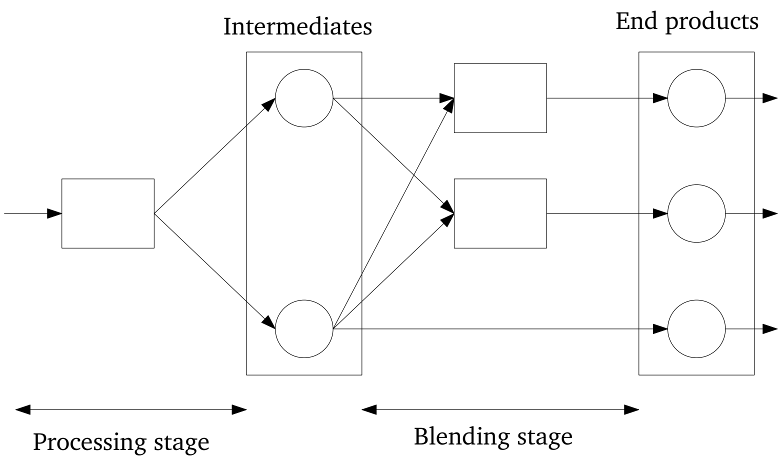

Figure 2.1: An example production system

The problem we address in this study involves the selection of (i) a set of intermediates to be stocked from a given set of potential intermediates, and (ii) end product recipes which specify how those intermediates are blended into end products. The selection of intermediates and end product recipes is associated with a set of cost factors and constraints. The total operational cost is composed of material procurement costs, processing costs, storage costs associated with selected intermediates; and blending costs associated with end products. There are two basic constraint sets. First, the compositions of end products, which are defined by their product recipes, must comply with a specified set of quality requirements in order to guarantee the conformity of end products. The composition of an end product characterizes all types of attributes associated with it. However, often it is sufficient to consider a subset of those attributes when dealing with the quality requirements of end products. Here, we refer to those attributes as quality parameters. The quality requirement regarding a particular quality parameter (e.g. protein or fat concentration) states that the relevant parameter must be within a given range defined by a minimum and a maximum. Second, available production and storage capacities must be sufficient to put the selected intermediates and end product recipes to use. That is, given a set of selected intermediates and end product recipes, it must be possible to produce the necessary amount of intermediates and to blend them into end products to satisfy the demand, and the storage facilities must be sufficient to stock production lots.

The processing stage is characterized by processing and setup times/costs associated with each intermediate. In order to avoid high setup costs and down times, long processing runs and/or a limited number of intermediates are preferred. Production operations are scheduled following the common cycle scheduling policy (Hanssmann, 1962). This approach is widely used in industry due to its simplicity and adaptability and has been proven to produce optimal or near-optimal schedules in many practical situations especially when products are similar in terms of their cost structure and demand rates, and setup times are relatively short (Jones and Inman, 1989). In a common cycle schedule, one lot of each product is produced in each production cycle and the cycle time is identical for each product (in our case selected intermediate). If the usage rates of selected intermediates were known in advance, then the optimal cycle time could easily be determined following to the common cycle policy. However, in our case, the usage rates depend on the decisions regarding the set of selected intermediates and end product recipes. Hence, rather than optimizing the cycle time we aim at finding the optimal set of selected intermediates and end product recipes for a given cycle time. Due to the perishable nature of food products, cycle times are rather short in food processing industry. Furthermore, cycle times are usually not just arbitrary intervals but integer multiples of an applicable time period such as a shift or a day. Thus, in case it is needed, the model can be solved for a limited set of applicable cycle lengths.

As discussed previously, the product variety at the end product level often makes it impossible to store all end products. Because of this reason, blending operations run on a daily basis following end product demand. The blending stage usually involves very standardized operations. Hence we assume a constant blending rate for all end products. The setup operations in this stage are minor and are assumed to be negligible.

The selected intermediates are stored between the two production stages in a number of storage units (e.g. silos or tanks) which are identical in terms of their volume. The limitations on the storage capacities are rather restrictive in the food processing industry since only a single type of intermediate can be stored in a storage unit (Akkerman et al, 2007). The customer preferences and demands change gradually over time, and consequently, selected intermediates and end product recipes are usually revised to correct for those periodically. We assume that demand is stable within those revision intervals.

2.3 Related literature and positioning

The first example of the blending problem is the famous diet problem of Stigler (1945) where a minimum cost diet is determined subject to a set of dietary allowances. Following the line of this problem a large body of literature has emerged addressing blending problems particularly in the petrochemical industry and the agricultural industry. Most of this work has been concentrated on stand-alone blending problems which usually concern the determination of a minimum cost blend or a recipe while respecting a set of quality related constraints. However, in processing systems, the production and the storage operations are tightly coupled with product recipes and demands which together determine the consumption rates of the ingredients to be used in processing the blends. Crama et al (2001) classify blending problems into three basic categories based on the degree of the integration of the blending problem with production and storage operations: (i) design problems where the blending operations are considered in isolation, (ii) long- or medium-term planning problems where the blending operations are integrated in the long- or medium-term (master) planning, and (iii) short term planning and scheduling problems where the blending problem is a part of everyday operations. The problem under consideration in this study falls into the category of medium-term planning problems. Here we briefly review some of the work in this domain.

Glen (1980) develops a method for the beef cattle feedlot operations to determine the rations to feed animals. His method gradually changes the rations over time in order to obtain a specified liveweight at the minimum cost. Steuer (1984) studies sausage blending problems which concern the optimization of meat blends to produce sausages under a set of quality constraints. Taube-Netto (1996) presents an integrated planning model for poultry production which encompasses, among other aspects, the formulation of feed to be used over the planning horizon. In the aforementioned examples, the processing and blending operations of the feedstuffs are not integrated into the overall production planning problem.

Williams and Redwood (1974) propose a multi-period blending model for a company that refines and blends different types of raw oils to produce a number of brand oils. Their model decides upon the purchasing and production quantities for each time period considering the price fluctuations of raw oils. Rutten (1993) develops a hierarchical approach for the operational planning of a dairy firm. He considers the planning problem at the operational planning level and decomposes it into smaller problems each of which can be solved in reasonable computational times. However, these studies do not consider the economies of scale resulting from the setup costs/times.

Karmarkar and Rajaram (2001) study the joint production and blending problem. They propose a general mixed integer non-linear program (MINLP) and a Lagrangean heuristic to solve the problem. Their work is substantial since they jointly optimize the lot sizes and end product recipes. However, they consider only a single quality parameter and use a cost function to penalize the nonconformity of end product. Furthermore, they assume uncapacitated production and storage.

Our study is closely related to the work of Akkerman et al (2010) where a flour manufacturing system is considered. They study a system where a limited number of grains are milled and blended into various types of flour products. They propose a mixed integer linear program (MILP) to determine the recipes of flour products minimizing total milling and blending costs. Their approach also accounts for the option of using selected intermediates directly as end products. They do not explicitly consider the production and storage capacities. However, they approximate these limitations by using an upper bound on the number of intermediates to be selected. They mention that it is logical to limit the number of intermediates since the opposite would require large setup times and a huge storage capacity. In this study, we build on the model provided by Akkerman et al (2010) and extend their study by explicitly incorporating the capacity limitations and costs on production and storage operations.

2.4 Model formulation

In this section, we present a mathematical model for the intermediate selection and blending problem. We first provide the notation used in the rest of the paper. Then we outline the objective function and the constraints characterizing the problem.

2.4.1 Notation

Consider a food processing system producing a set of end products J. These end products can be produced by using a set of intermediates I. Intermediates and end products are characterized by their compositions in terms of a set of ingredients K. We refer to the proportions of those ingredients as quality parameters. The quality parameters of intermediates are known whereas they are defined on minimum and maximum levels for end products. The end product recipes should comply with those bounds.

We are given the quality specifications

q_{ik} = quality parameter $k \in K$ of intermediate $i \in I$ (%)
$q_{jk}^{\min}$ = minimum quality parameter $k \in K$ of end product $j \in J$ (%)
$q_{jk}^{\max}$ = maximum quality parameter $k \in K$ of end product $j \in J$ (%)

demand and process characteristics

d_j = demand rate of end product $j \in J$ (tons/day)
s_i = setup time of intermediate $i \in I$ (days)
p_i = processing rate of intermediate $i \in I$ (tons/day)
p^{b} = blending rate of end products (tons/day)
N = number of available storage units
V = capacity of each storage unit (tons)
π = cycle time (days)

and cost parameters

a_i = setup cost of intermediate $i \in I$ (Euros)
c_i = processing (and material) cost of intermediate $i \in I$ (Euros/ton)
c^{b} = blending cost of end products (Euros/ton)
h_i = holding cost of intermediate $i \in I$ (Euros/ton day).

In order to specify the basic intermediates to be used and corresponding end product recipes we define the variables

x_{ij} = fraction of end product $j \in J$ supplied by intermediate $i \in I_j$

where $I_j \subset I$ is the set of intermediates which can be used in producing end product j,

$$y_i = \begin{cases} 1, & \text{if intermediate } i \in I \text{ is selected as a basic intermediate} \\ 0, & \text{otherwise} \end{cases}$$

and

$$v_{ij} = \begin{cases} 1, & \text{if intermediate } i \in I_j^* \text{ is used directly as end product } j \in J \\ 0, & \text{otherwise} \end{cases}$$

where $I_j^* \subset I_j$ is the set of intermediates which comply with all quality specifications of end product j, i.e.

$$I_j^* = \{i \in I_j \mid q_{jk}^{\min} \leqslant q_{ik} \leqslant q_{jk}^{\max}, \forall k \in K\}.$$

For notational simplicity, we also introduce the expressions

w_i = the consumption rate of intermediate $i \in I$ (tons/day)

such that,

$$w_i = \sum_{j \in J} d_j x_{ij} \quad \forall i \in I \tag{2.1}$$

and

$$z_j = \begin{cases} 1, & \text{if end product } j \in J \text{ is produced with blending operations} \\ 0, & \text{otherwise} \end{cases}$$

such that,

$$z_j = 1 - \sum_{i \in I_j} v_{ij} \quad \forall j \in J. \tag{2.2}$$

Notice that, the domain of z_j can be verified since v_{ij} equals 1 for at most one intermediate. This will further be clarified in the constraints.

2.4.2 Objective function

The objective is to minimize the daily total costs which is comprised of cost components associated with setup, processing and storage of intermediates; and blending of end products. Setup costs are relevant to those intermediates which are selected as basic intermediates. Since processing operations are carried out following a common cycle schedule, in each cycle a setup is initiated for every basic intermediate. Thus, cost incurred in a single cycle equals $\sum_{i\in I} a_i y_i$. To obtain the setup cost per day, the cost per cycle is divided by the cycle time. Processing costs involve the material and operational costs of processing operations, and they are incurred for all basic intermediates in proportion to their consumption rates. Hence, daily processing cost can be expressed as $\sum_{i\in I} c_i w_i$. It is important to note that processing cost, as a combination of material and operational costs, is usually the largest cost component of the total costs in food process industries. Storage costs depend on the average inventory levels of intermediates. The processing of an intermediate, say intermediate i, starts when the inventory drops down to zero, and stops when reaches up to $\pi w_i(1 - w_i/p_i)$. Because both production and consumption rates are assumed to be constant, the average inventory equals half of the maximum inventory level. Thus, $\sum_{i\in I} 0.5 h_i \pi w_i(1 - w_i/p_i)$ gives the daily storage cost. Blending costs are incurred for end products which go through the blending operation in proportion to their demand rates. Hence, the daily blending cost equals $c^{\mathrm{b}} \sum_{j\in J} d_j z_j$. The following expression, therefore, provides daily total costs.

$$\frac{1}{\pi}\sum_{i\in I} a_i y_i + \sum_{i\in I} c_i w_i + \sum_{i\in I} \frac{1}{2} h_i \pi w_i \left(1 - \frac{w_i}{p_i}\right) + c^{\mathrm{b}} \sum_{j\in J} d_j z_j. \tag{2.3}$$

2.4.3 Constraints

The capacitated intermediate selection and blending problem involves a variety of constraints regarding the conservation and quality requirements of end product recipes, and capacity limitations on processing, storage and blending operations. These constraints are articulated in this subsection.

Recipe conservation constraints. For each end product j, the fractions x_{ij} defining the contribution of each intermediate i into end product j must sum up to 1 in order to specify a complete recipe:

$$\sum_{i\in I_j} x_{ij} = 1 \quad \forall j \in J. \tag{2.4}$$

The decision on whether intermediate i is selected to be used in one or more end product recipes is indicated by the binary decision variable y_i. Hence, intermediate i cannot take place in any end product recipe as long as y_i equals 0:

$$x_{ij} \leqslant y_i \quad \forall i \in I_j, \forall j \in J. \tag{2.5}$$

If end product j is directly supplied as intermediate i then its contribution in the associated recipe (in percentage) must equal 1 (i.e. %100):

$$v_{ij} \leqslant x_{ij} \quad \forall i \in I_j^*, \forall j \in J. \tag{2.6}$$

Notice that Eq. (2.6) together with Eq. (2.4) guarantees that $\sum_{i \in I_j^*} v_{ij} \in \{0,1\}$, and hence $z_j \in \{0,1\}$.

Quality constraints. Quality constraints guarantee that recipes comply with the quality requirements of end products. That is, each quality specification k of end product j, as the weighted average of the specifications of the intermediates take place in the corresponding recipe, must be between the pre-specified minimum and maximum quality parameters:

$$q_{jk}^{\min} \leqslant \sum_{i \in I_j} q_{ik} x_{ik} \leqslant q_{jk}^{\max} \quad \forall j \in J, \forall k \in K. \tag{2.7}$$

Processing capacity constraints. Processing capacity constraints make sure that there is enough time for the setup and the production operations of the selected intermediates within the given cycle length. This can be guaranteed by

$$\sum_{i \in I} \left\{ s_i y_i + \pi \frac{w_i}{p_i} \right\} \leqslant \pi \tag{2.8}$$

where the terms in the summation stand for the total setup time and the total processing time associated with the selected intermediates respectively. Notice that, w_i equals 0 for those intermediates that are not selected (see Eq. (2.1) and Eq. (2.5)).

Storage capacity constraints. Storage capacity constraints limit intermediate inventory levels. More specifically, there are N storage silos available each with V tons of capacity. Because the form of storage is homogeneous, the number of storage units constitutes an upper bound on the number of intermediates. However, it is also possible to assign multiple storage units to a particular intermediate. Henceforth, the number of storage silos assigned to an intermediate bounds the maximum inventory level of that intermediate.

The maximum inventory level of an intermediate depends on the production mode as discussed in forming the objective function. Following the same reasoning, we can write the storage constraints as

$$\sum_{i\in I}\left\lceil \frac{\pi w_i\left(1-\frac{w_i}{p_i}\right)}{V}\right\rceil \leqslant N. \tag{2.9}$$

Blending capacity constraints. Blending capacity constraints limit the extent of the daily blending operations. The daily blending rate is given as p^{b}. The total daily blending volume is the sum of the demands associated with those end products which undergo blending operations as indicated by the binary variable z_j. Hence, the daily blending capacity constraint is expressed as

$$\sum_{j\in J} d_j z_j \leqslant p^{\mathrm{b}}. \tag{2.10}$$

The mathematical formulation provided so far involves non-linear expressions both in the objective function and constraints. In particular, both storage costs and constraints are non-linear (see Eq. (2.3) and Eq. (2.9)). These expressions are difficult to handle with general purpose mathematical programming solvers. In Appendix 2.A, we provide a linearization scheme for those expressions which enables us to express the model as a MILP. Also, in Appendix 2.B, we provide some upper bounds on the consumption rates of potential intermediates which can be used to strengthen the formulation.

2.5 Numerical study

We implement our approach on a data set collected from a medium-sized flour manufacturer that supplies flour products to bakeries and industrial manufacturers. The main processing operation in flour manufacturing is the milling process where the grains are ground between successive sets of mill stones or rollers to produce different types of intermediate flour products. These flour products can be used directly as end products to satisfy demands. Alternatively, they can be blended into end products following to a blending operation where they are dispersed within each other and homogenized. Consequently, the decision problem is to select those flour products to be stocked, and to determine the recipes of end products specifying how they are blended.

The production system under consideration involves 76 potential intermediates and 45 end products with 9 different quality parameters. The flour mill can process flour products with a capacity around 350 tons/day. This does not include the setup times which are around 30 minutes per changeover. The blender mixes flour products with an average capacity of 200 tons/day. There are 18 storage silos each of which can store around 50 tons of material. The total demand sums up to 220 tons/day. However, the demands vary substantially between different end products. We do not provide the cost figures here for the sake of confidentiality. It is important to note that, as it is in most process industries, the material procurement costs (which are expressed in processing costs in the formulation) are dominant with respect to other operational costs. However, while good purchasing is pivotal, minimizing the non-procurement costs is important to stay competitive since profit margins are rather small in the food processing industry.

We analyze the case in a constructive manner by using several, increasingly comprehensive, scenarios, thereby illustrating the effect of the different constraints and cost factors. In Scenario 1, we consider the blending problem in isolation, ignoring all types of capacity limitations. This scenario establishes a benchmark to compare the following scenarios with. In Scenario 2, we add the production capacity limitations and setups costs to the problem showing us how these affect the selection of intermediates and end product recipes. In Scenario 3, we integrate the storage costs into the problem, demonstrating the trade-off between setup costs and storage costs. In Scenario 4, we finally add the storage capacity limitation, thereby considering all relevant costs and capacity limitations. This scenario reflects the actual production environment addressed in this study. In Scenario 5, we again study the complete problem, but change the production setup of the case company to look at possible ways to increase efficiency.

In each scenario, we communicate the daily costs and capacity utilization levels of production and storage operations, and provide some basic information regarding the selection of intermediates and end product recipes. More specifically, we report:

1. Cycle time (CT): π
2. Total costs (ToC): $1/\pi \sum_{i \in I} a_i y_i + \sum_{i \in I} c_i w_i + \sum_{i \in I} 1/2 h_i \pi w_i (1 - w_i/p_i) + c^{\mathrm{b}} \sum_{j \in J} d_j z_j$
3. Processing costs (PrC): $\sum_{i \in I} c_i w_i$
4. Setup costs (SeC): $1/\pi \sum_{i \in I} a_i y_i$
5. Blending costs (BlC): $c^{\mathrm{b}} \sum_{j \in J} d_j z_j$
6. Storage costs (StC): $\sum_{i \in I} 1/2 h_i \pi w_i (1 - w_i/p_i)$

7. Processing utilization (PrU): $1/\pi \sum_{i \in I}\{s_i y_i + \pi w_i/p_i\}$
8. Blending utilization (BlU): $\sum_{j \in J} d_j z_j / p^{\mathrm{b}}$
9. Storage utilization (StU): $\sum_{i \in I} \lceil \pi w_i(1 - w_i/p_i)/V \rceil / N$
10. Number of selected intermediates (#SI): $\sum_{i \in I} y_i$
11. Number of end products directly supplied from intermediate stocks (i.e. end products with a single intermediate in their recipes) (#EPFS): $|J| - \sum_{j \in J} z_j$

Notice that, we define the utilization levels as the ratio of the engaged capacity to the available capacity. As such, the storage utilization relates to the percentage of storage units in use, rather than utilization of each individual storage unit.

2.5.1 Scenario 1

We start our analysis with the blending problem in isolation. That is, we minimize the sum of processing and blending costs subject to the quality constraints of end products. Thus, we assume that the production and storage capacities are both infinite and we neglect the setup and storage costs. Notice that, the optimal solution of this problem is independent of the cycle time, and it sets a lower bound on the costs for the original problem. The results are given in Table 2.5.1.

	Costs (Euros/day)					Utilization (%)			Recipes	
CT	ToC	PrC	SeC	BlC	StC	PrU	BlU	StU	#SI	#EPFS
1	31202	31129	-	73	-	-	-	-	30	32

Table 2.1: Optimal solution - Scenario 1

We observe that the daily total costs are minimized at 31202. Only a very small portion of this cost originates from the blending operations. The model selects 30 out of 76 intermediates to be stocked and 32 out of 45 end products to be supplied from stock. One could expect that supplying all end products from stock yields a lower total cost by preventing any blending costs. Yet, the optimal solution suggests that 13 of 45 end products should undergo blending operations. This shows that the unit production costs of those end products which are not directly supplied from stock are smaller when they are blended from a number of intermediates despite the additional blending costs. In other words, the intermediates which comply with the quality requirements of those end products possess larger unit processing costs than the unit processing cost of the optimal blend plus the blending costs. Another result is that 32 end products are supplied from stock although only 30 intermediates are selected to be stocked. This means that some selected intermediates comply with the quality requirements of multiple end products. Notice that this scenario reflects the minimum attainable combination of processing and blending costs. In the following scenarios, we analyze how additional costs and capacity limitations add on this cost figure.

2.5.2 Scenario 2

In this scenario, we integrate production capacities (i.e. processing rates and setup times) and related costs (i.e. setup costs) into the problem considered in Scenario 1 while still neglecting the storage capacity limitation and storage costs. Note that the optimal solution of the problem is now dependent on the cycle time. In Table 2.2, we therefore report the optimal solutions of the problem for cycle times of 1 to 10 days.

	Costs (Euros/day)					Utilization (%)			Recipes	
CT	ToC	PrC	SeC	BlC	StC	PrU	BlU	StU	#SI	#EPFS
1	31875	31215	531	129	-	0.88	0.32	-	11	21
2	31585	31178	290	117	-	0.77	0.29	-	12	21
3	31480	31165	224	91	-	0.74	0.23	-	14	24
4	31423	31160	174	89	-	0.71	0.22	-	15	25
5	31386	31151	146	89	-	0.71	0.22	-	16	25
6	31363	31153	124	86	-	0.70	0.21	-	16	25
7	31346	31153	107	86	-	0.69	0.21	-	16	25
8	31331	31141	102	88	-	0.70	0.22	-	17	24
9	31319	31142	90	87	-	0.69	0.22	-	17	25
10	31311	31132	92	87	-	0.72	0.22	-	19	25

Table 2.2: Optimal solution - Scenario 2

The results show that both the cost figures and the selection of intermediates are quite different from the ones in Scenario 1. We notice a sharp decrease in the number of selected intermediates compared to Scenario 1. This leads to higher processing and blending costs which, together with setup costs, significantly increase the daily total cost. Notice that the optimal daily total cost is decreasing on cycle time since we do not consider storage costs. We also observe that neither the production nor the blending capacity is binding. For the cycle times considered the utilization of the production and blending do not reach 100%.

It is important to remark that the selection of the intermediates is also affected by the cycle time. A setup cost is incurred for each selected intermediate in every production cycle. Thus, the daily total setup cost is decreasing on the cycle time and increasing on the sum of the individual setup times of selected intermediates. In this sense, having a larger number of selected intermediates may lead to higher setup costs. The results clearly show that this effect is dominated by the cost reduction due to the increasing cycle time. On the other hand, having a larger number of intermediates may also result in lower processing and/or blending costs by bringing more options to supply end products. We can detect these effects in Table 2.2. The results show that increasing cycle time leads to a larger number of selected intermediates, and thus, reduce processing and blending costs.

2.5.3 Scenario 3

In this scenario, we integrate the storage costs into the problem considered in Scenario 2 while still neglecting the storage capacity limitation. In Table 2.3 we report the optimal solutions of the problem for cycle times of 1 to 10 days.

	Costs (Euros/day)					Utilization (%)			Recipes	
CT	ToC	PrC	SeC	BlC	StC	PrU	BlU	StU	#SI	#EPFS
1	31892	31215	531	129	17	0.88	0.32	-	11	21
2	31623	31178	290	117	38	0.77	0.29	-	12	21
3	31539	31165	224	91	59	0.73	0.23	-	14	24
4	31503	31165	168	91	79	0.71	0.23	-	14	24
5	31485	31151	146	89	99	0.71	0.22	-	16	25
6	31482	31153	124	86	119	0.70	0.21	-	16	25
7	31485	31153	107	86	139	0.69	0.21	-	16	25
8	31490	31142	102	87	159	0.70	0.22	-	17	25
9	31498	31142	90	87	179	0.69	0.22	-	17	25
10	31509	31142	81	87	199	0.69	0.22	-	17	25

Table 2.3: Optimal solution - Scenario 3

The difference between the cost figures in Scenario 2 and Scenario 3 demonstrates the effects of storage costs. We observe that the daily setup cost is decreasing whereas the daily storage cost is increasing on cycle time. As a consequence, the optimal daily total cost is no longer decreasing on cycle time. The minimum daily total cost 31482 is achieved when the cycle time is 6 days. The optimal daily total cost is higher for cycle times shorter than 6 days due to larger setup costs, and for cycle times longer than 6 days due to larger storage costs.

The effect of holding costs on the selection of intermediates is only visible for longer cycle times where the magnitude of storage costs is rather large. Consider the cycle time of 10 days. In Scenario 2, the model selects 19 intermediates which can supply 25 end products directly from stock. In Scenario 3, however, the model selects 17 intermediates which can also supply 25 end products directly from stock. The difference between those figures can be explained as follows. The total consumption rate of selected intermediates equals the total demand rate, and it is allocated between the selected intermediates following the end product recipes. The average inventory level of a selected intermediate is increasing on intermediate's production rate whereas it is concave on intermediate's consumption rate. Thus, with other things held constant, the total holding cost would be lower when the production rates and holding costs of the selected intermediates are lower and/or the number of selected intermediates is smaller. The reduction in the number of intermediates also leads to a slight increase in the processing costs, demonstrating that more expensive raw materials are required to produce more flexible intermediates.

2.5.4 Scenario 4

In this scenario, we integrate the storage capacity limitation into the problem considered in Scenario 3. Thus, we consider all types of capacity limitations and costs and investigate the actual real-life problem. In this particular case, there are 18 storage units available. In Table 2.4 we report the optimal solutions of the problem for cycle times of 1 to 6 days. We do not consider cycle times longer than 6 days because there is no feasible solution for those with the given storage capacity limitation.

	Costs (Euros/day)					Utilization (%)			Recipes	
CT	ToC	PrC	SeC	BlC	StC	PrU	BlU	StU	#SI	#EPFS
1	31892	31215	531	129	17	0.88	0.32	0.61	11	21
2	31623	31178	290	117	38	0.77	0.29	0.78	12	21
3	31539	31165	224	91	59	0.74	0.23	1.00	14	24
4	31577	31215	155	128	79	0.73	0.32	1.00	13	21
5	31891	31386	93	327	85	0.82	0.82	1.00	9	11
6	32451	31959	54	346	92	0.90	0.87	1.00	7	12

Table 2.4: Optimal solution - Scenario 4

The results show that minimum daily total cost is achieved when the cycle time is 3 days. We observe that the storage capacity limitation significantly affects the optimal cost structure and the selection of intermediates and end product recipes. Notice that, in general, a longer cycle time leads to higher average inventory levels. Thus storage capacity limitation is more restrictive when the cycle time is longer. It is possible to observe this by comparing the results reported in Table 2.3 and Table 2.4. The storage capacity limitation leads to large differences in daily total costs especially when the cycle time is longer than 2 days. The magnitude of this effect gradually increases with the cycle time and eventually results in an infeasible problem for cycle times longer than 6 days. The same effect can be observed on the storage utilization. For cycle times longer than 2 days, the utilization of storage units reaches 100%, and the storage capacity becomes binding. Hence, for those cycle times the cost difference between Scenario 3 and Scenario 4 originates from the limited storage capacity.

We also observe that the storage limitations significantly change the structure of the optimal set of selected intermediates and the end product recipes. In particular, for those cycle times where the storage capacity is binding, the model selects a smaller number of intermediates, preferably the ones with lower production rates, in order to reduce the stock levels. This, however, increases processing and blending costs because the set of selected intermediates and end product recipes further move away from the optimal ones.

2.5.5 Scenario 5

In the previous scenarios, we have observed that the storage capacity limitation is the most critical one among other limitations for the particular example considered in this numerical study. Consequently, in this scenario, we focus on cost reductions that can be achieved by altering the storage capacity. We conduct our analysis as follows. First, for each cycle time, we find a critical storage capacity level which is large enough to provide the optimal daily total cost that can be achieved when there is no storage limitation. Since inventory levels gradually increase with cycle time, we expect critical storage capacity levels to be higher for longer cycle times. These levels can easily be found by solving the problem without storage capacity limitations, as in Scenario 3, and checking how many storage units are being used following the optimal solution. Notice that the storage capacity constraint is binding only when the storage capacity is below those critical levels, and if so then having extra storage capacity could reduce the daily total cost. Secondly, we solve the problem for all storage capacity levels up to critical ones so as to find the added value of expanding the storage capacity.

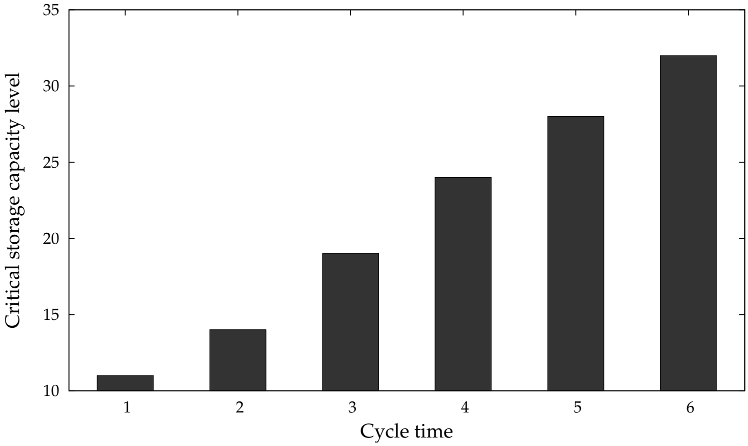

Figure 2.2: Critical storage capacity levels for cycle times of one to six days

We know, from Scenario 3, that the optimal cycle time is 6 days for the problem without storage capacity limitations. This cycle time can be regarded as an upper bound for the optimal cycle time when storage capacity is limited. Thus we limit our analysis to cycle times from 1 to 6 days. In Figure 2.2, we report those critical resource levels. As expected, we observe that the critical storage levels are increasing on cycle time. The critical storage capacity levels reflect the maximum number of storage units that could possibly be needed when a given cycle time is employed. Up to this level, additional storage units lead to lower costs, but, there is no added value of expanding the storage capacity beyond the critical level as long as the same cycle time is employed. Nevertheless, from Scenario 3 we know that the daily total cost can still be improved by employing a longer cycle time (not more than 6 days).

In what follows, we solve the problem for cycle times of 1 to 6 days and for all feasible storage capacity levels up to the critical ones so as to find the added value of the extra storage capacity. In previous scenarios we have already demonstrated the effects of the storage capacity limitation on individual cost components, utilization rates, and the selection of intermediates and end product recipes. Thus, in this scenario we only communicate the optimal daily total costs. Figure 2.3 illustrates the minimum daily total costs that can be achieved with respect to the available number of storage units – also including the cycle time linked to these solutions.

We observe that different cycle times are optimal for different storage capacities.

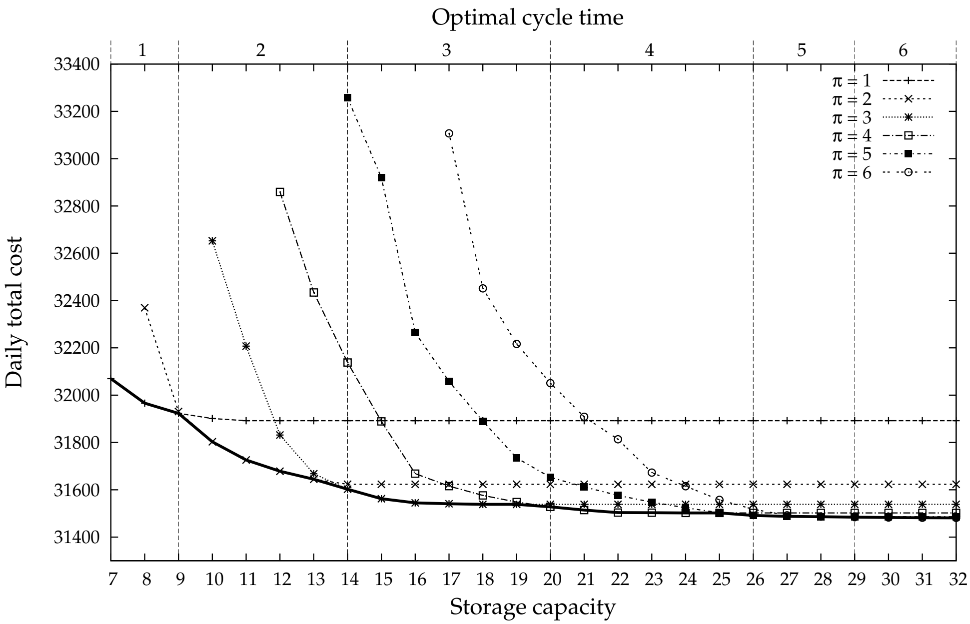

Figure 2.3: Optimal daily total costs - Scenario 5

For example, when cycle time is 3 days, the critical storage capacity level equals 19 storage units. Further increasing the storage capacity does not lead to a reduction of cost as long as the cycle time remains the same. However, increasing the available number of storage units to 20, while also increasing the cycle time to 4 days opens up the possibility to reduce costs, since the storage capacity of 20 units is less than the critical storage capacity level for the cycle time of 4 days. Hence, further increasing the storage capacity leads to a lower daily total cost until we reach a storage capacity of 25. After this, we would again need to increase the cycle time to enable further cost reductions. Notice that the cycle time that minimizes the daily total cost tends to increase as the storage capacity gets larger. Nevertheless, from Scenario 3, we know that the maximum cycle time that would be used equals 6 days. Hence, it would never be necessary to use more than 32 storage units as it is the critical storage level for the cycle time of 6 days.

We can also investigate the added value of extra storage capacity by looking at the minimum daily total cost for each storage capacity level over the cycle times. These costs are illustrated by the down-most bold line in Figure 2.3. We know that the daily total cost tends to decrease as the storage capacity expands up to 32 storage units and then levels off. However, we observe that this trend is not steady over the storage capacity. That is, the cost reduction due to an extra storage unit is not decreasing on the number of available storage units. This is mainly because when an additional storage unit increases the storage capacity to exceed a critical level, it brings up the possibility of reducing the cost also by altering the cycle time which is not possible otherwise.

In the actual storage setting of the case company, 18 storage units are used. We know now that expanding the storage capacity to 32 units will reduce the daily total cost – but only if the cycle time is increased simultaneously. Minor, more realistic, increases in the storage capacity should also be considered in combination with changes in the cycle time. Adding one storage unit has almost no effect on costs whereas adding two or three storage units in combination with an increase of the cycle time to 4 would have a significant effect.

2.6 Conclusions and extensions

In this study, we addressed a capacitated intermediate product selection and blending problem encountered in the food processing industry. The problem involves the selection of a set of intermediates and end product recipes characterizing how those selected intermediates are blended into end products to minimize the total operational costs under capacity limitations. We developed a comprehensive mixed integer linear model for the problem. We applied the model to a data set collected from a real-life company and we analyzed the problem under several scenarios to better understand the trade-offs between capacity limitations and costs. For the particular case considered in our numerical study, we observed that the production and the blending capacities are not binding for the case whereas the storage capacity is. Consequently, we investigated possible cost reductions that can be achieved by altering the storage capacity. We showed that the cost reduction due to an extra storage unit is not decreasing on the number of available storage units, mainly due to the use of different cycle times. This suggests that a careful investigation is required when deciding upon an expansion of the storage capacity.

In general, this study demonstrated important product-process interactions in the process industries, where the decisions on the selection of intermediates and the configuration of end product recipes are affected by the capacity limitations and the costs associated with production and storage operations. In this context, the problem addressed in this study can be regarded an extension of the production lot scheduling problem with integrated design decisions. The conventional production lot scheduling problems try to balance the trade-off between the fixed production setup costs and inventory holding costs. We observed that this trade-off is affected by integrated design decisions, since excessive setup and holding costs can be eliminated by reducing the number of intermediates, however, in expense of additional blending costs. We also analyzed how capacity limitations affect the selection of intermediates and end product recipes. We saw that these limitations interact with each other, and based on their magnitude, one or more of those limitations could be binding at the same time. Especially the planning of the production operations within the capacitated situation (in this paper represented by the selection of a certain cycle time) had a significant effect on the selected intermediate products, the amount of storage units this requires, and the total costs. It should be noted that we mainly concentrated on the storage capacity in our numerical study because it appeared to be the most critical parameter for the case example under consideration. Nevertheless, there are many other parameters, such as production capacities, and setup and holding costs which could also be influential. The model presented in this paper can be used to gain insight in the complex interactions between product design, process design, and operational planning. Also, an analysis as the one conducted in this paper, could be useful for evaluating alternative design and/or expansion decisions.

This research is particularly aimed at the food processing industry. Nevertheless, it encompasses characteristics, such as limitations on production and storage capacities, which are very common in many other processing systems. Therefore, the proposed model can also be adapted to other processing systems with some simple modifications. For instance, in some processing systems, production lots go through a series of quality checks before they are used. This could easily be covered by the proposed model by replacing the expression of maximum inventory levels $\pi w_i(1 - w_i/p_i)$) with πw_i in the objective function and the storage capacity constraints. This would also simplify the linarization of storage capacity constraints. There are several directions we leave aside for further research. We analyzed the problem assuming a common cycle scheduling policy. Although it is widely used in practice, under certain circumstances this policy may perform badly. Hence, the same problem can be considered under more sophisticated scheduling policies. We assumed that all storage units are identical, and they must be assigned to certain intermediates. Violating these assumptions require significant modifications in our approach. However, in some cases, storage units may possess different characteristics, and it may be possible to switch storage units between intermediates. Thus, it would be interesting to analyze the problem while relaxing these assumptions.

Appendix 2.A Piecewise linear approximation of storage capacities and costs

The mathematical formulation provided in Section 2.4 involves non-linearities in the objective function and constraints. The non-linearity arises due to the expression of the maximum inventory level of intermediates which is a quadratic polynomial. The expression appears in the objective function (see Eq. (2.3)) and the storage capacity constraints (see Eq. (2.9)). Here, we provide a piecewise linear approximation for this expression. Let us define

$$f_i(w_i) = \pi w_i \left(1 - \frac{w_i}{p_i}\right). \tag{2.11}$$

For any intermediate i, the consumption rate w_i is non-negative, and it cannot exceed the production rate p_i (see Eq. (2.8)). Hence, we analyze $f_i(\cdot)$ in the domain $[0, p_i]$, where it is concave, equals 0 at $w_i = 0$ and $w_i = p_i$, and reaches its maximum at $w_i = p_i/2$ (see Figure 2.4).

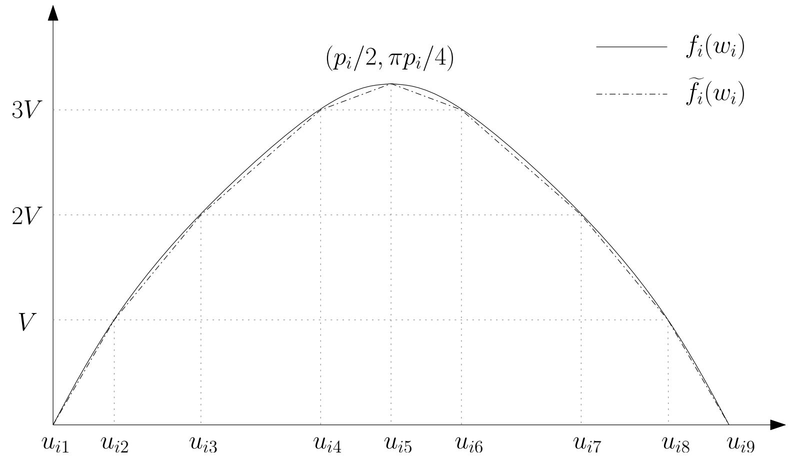

Figure 2.4: Approximation of $f_i(\cdot)$

The proposed approximation scheme is based on setting breakpoints of the piecewise approximation to exact values of $f_i(w_i)$ corresponding to integer multiples of the storage unit capacity V. In order to account for the exact maximum inventory level and extreme values of the domain of w_i, we also use breakpoints at $w_i = 0$, $w_i = p_i/2$, and $w_i = p_i$. This approach guarantees the feasibility of any storage unit assignment while, in general, underestimating maximum inventory levels and hence storage costs.

The number of linear segments that must be used for intermediate i depends on the production rate p_i, the capacity of storage units V, and the length of the planning horizon π. We denote the set of linear segments for intermediate i by L_i. Each linear segment $l \in L_i$ is bounded below and above by two breakpoints denoted by u_{il-1} and u_{il}. Notice that these values can easily be pre-computed. Figure 2.4 depicts a possible realization of $f_i(\cdot)$ and the corresponding approximation function denoted by $\widetilde{f}_i(\cdot)$.

The approximation scheme requires the usage a set of new variables and constraints. Let α_{il} and β_{il} be the intercept and the slope of the l'th linear segment of $\widetilde{f}_i$. Now, we introduce variables:

$$\delta_{il} \quad = \begin{cases} 1, & \text{if } u_{il-1} < w_i \leqslant u_{il} \\ 0, & \text{otherwise} \end{cases}$$

and

$$\mu_{il} \quad = \begin{cases} w_i, & \text{if } u_{il-1} < w_i \leqslant u_{il} \\ 0, & \text{otherwise.} \end{cases}$$

It should be obvious that δ_{il} and μ_{il} represent the active segment and the corresponding consumption rate for intermediate i. In order to guarantee that these variables take appropriate values, we use the constraints:

$$u_{il-1}\delta_{il} < \mu_{il} \leqslant u_{il}\delta_{il} \quad \forall l \in L_i, \forall i \in I \tag{2.12}$$

and

$$\sum_{l \in L_i} \delta_{il} = 1 \quad \forall i \in I \tag{2.13}$$

$$\sum_{l \in L_i} \mu_{il} = w_i \quad \forall i \in I. \tag{2.14}$$

We can now re-write the objective function in Eq. (2.3) by replacing the exact expression of the maximum inventory levels given in Eq. (2.11) with the following approximate one:

$$\widetilde{f}_i(w_i) = \sum_{l \in L_i} \alpha_{il}\delta_{il} + \beta_{il}\mu_{il}. \tag{2.15}$$

Next, we revise the storage capacity constraint. The number of storage tanks that must be assigned to intermediate i is uniquely defined for each linear segment $l \in L_i$.

Let us denote these values by r_{il}, such that, if $\delta_{il} = 1$, then r_{il} storage tanks are assigned to intermediate i. We can now re-write the storage capacity constraint given in Eq. (2.9) as

$$\sum_{i \in I} \sum_{l \in L_i} r_{il} \delta_{il} \leqslant N. \tag{2.16}$$

Appendix 2.B Upper bounds on the consumption rates of intermediates

We used a piecewise linear approximation scheme in order to ensure the linearity of the mathematical formulation. However, this necessitated the use of a new set binary variables. In this section, we provide a simple method to reduce the number of those variables, and thus, to reduce the computation time. The method is based on the idea of finding upper bounds on the consumption rates of potential intermediates by using a slightly modified version of the blending sub-problem. These bounds are then used to cut-off some of the binary variables associated with the assignment of storage units to intermediates.

Let us consider the blending sub-problem:

$$\min \sum_{j \in J} \sum_{i \in I_j} c_i x_{ij} \tag{2.17}$$

$$\sum_{i \in I_j} x_{ij} = 1 \quad \forall j \in J \tag{2.18}$$

$$q_{jk}^{\min} \leqslant \sum_{i \in I_j} q_{ik} x_{ik} \leqslant q_{jk}^{\max} \quad \forall j \in J, \forall k \in K \tag{2.19}$$

The blending sub-problem provides the optimal end product recipes when production and storage capacities are unlimited and the only cost to be considered is the processing costs of intermediates. Hence, the constraints of the blending sub-problem define all feasible end product recipes. Now, let us replace the objective function by

$$\max \quad w_i^{\max} = \sum_{j \in J} d_j x_{ij}. \tag{2.20}$$

It is clear that $w_i^{\max}$ is an upper bound on any feasible w_i of intermediate i in the original formulation since it is the maximum possible consumption rate in the uncapacitated problem. The modified version of the blending sub-problem is a simple linear program and can easily be solved for each intermediate. Notice that it is possible to find stronger bounds by using more sophisticated sub-problems following the same approach. However, we experienced considerable reductions in the computation time even with this simple sub-problem.

Once upper bounds on the consumption rates are computed, for each intermediate i, we can replace the set of linear segments L_i with a smaller one $\widetilde{L}_i$ which can be defined as follows:

$$\widetilde{L}_i = \{l \in L_i | u_{il} \leqslant w_i^{\max}\}. \tag{2.21}$$

Chapter 3

A discrete time formulation for batch processes with storage capacity and storage time limitations

This chapter is published as Kilic, Van Donk, and Wijngaard (2011), A discrete time formulation for batch processes with storage capacity and storage time limitations, Computers & Chemical Engineering, 35(4): 622 - 629.

Abstract. *This paper extends the conventional discrete time mixed integer linear programming (MILP) formulation for scheduling multiproduct/multipurpose batch processes by introducing storage capacity and storage time limitations. For this purpose, storage vessels are explicitly modeled on which material flows are defined, and storage capacity and storage time constraints are expressed. The approach is shown to be effective in modeling the scheduling problem in a variety of storage configurations such as single/multiple dedicated and multipurpose storage vessels. In a numerical study, cases where storage capacity and storage time limitations have significant impacts on scheduling production and storage operations are highlighted.*

3.1 Introduction

Following recent trends, the process industry has experienced growing logistical demands, growing variety in products, and more intense competition. The increase in the variety of products leads to the need for multiproduct and multipurpose batch processing systems offering high flexibility. On the one hand, such systems provide a means to meet the increasing demands with regard to the larger number of different products. On the other hand, they intensely affect planning and scheduling of operations by necessitating the use of limited equipment and resources to undertake a variety of tasks. In such systems, the impact of effective scheduling methods is significant. As a result, the inherent mathematical planning and scheduling problems have received considerable attention in the literature (Mendez et al, 2006).

There is a large variety of aspects that can be considered in developing scheduling models for batch processing systems. A detailed taxonomy can be found in Mendez et al (2006). One of these aspects is the specification of the constraints on storage operations. Storage operations lead to better utilization of resources by decoupling consecutive processes. Hence, limitations on the storage operations constrain the extent to which the upstream and downstream production units can be decoupled. There are two types of limitations with regard to storage operations: capacity constraints (in terms of the number and the size of storage vessels) and storage time constraints (in terms of the time materials deteriorate). Capacity constraints limit the amount of material that can be stored in storage vessels, whereas storage time constraints restrict the amount of time that materials can be stored in storage vessels e.g. before decaying. In presence of these constraints, inappropriate scheduling decisions may result in a high number of setups, blocking or starvation throughout the stages, waste of intermediates, and thus, degrade the overall system performance significantly.

In the literature, capacity and storage time constraints are reflected by intermediate storage policies (Mendez et al, 2006). Unlimited intermediate storage (UIS), finite intermediate storage (FIS), and no intermediate storage (NIS) policies are associated with capacity constraints, whereas, unlimited wait (UW), finite wait (FW), and zero wait (ZW) policies are associated with storage time constraints. Although the combinations of UIS, NIS, UW and ZW policies have been intensively investigated, the combination of FIS and FW policies is rather neglected in the literature (Sundaramoorthy and Maravelias, 2008). Nevertheless, the combination of FIS and FW policies are relevant and important in real-life settings. FIS policy is very common since batch plants hardly ever have unlimited storage. FW policy is essential for many industries (e.g. food, solvent, polymer and pharmaceuticals) which involve unstable/perishable intermediates/products that must be processed/shipped within a short time (Gupta and Karimi, 2003).

There is a variety of studies in the literature which aim at developing efficient mathematical scheduling models for multiproduct/multipurpose batch processes (see e.g. Kondili et al, 1993a; Schilling and Pantelides, 1996; Ierapetritou and Floudas, 1998). These models account for storage capacity limitations either by imposing an upper bound on the total inventory of each type of material, or by modeling and confining each storage vessel separately. Regardless of the approach taken, addressing storage time limitations together with storage capacity limitations is difficult due to the need of tracing the remaining storage life of materials in storage vessels. The problem gets even more demanding when storage vessels are not dedicated to certain materials, and batches of the same state are mixed and split through the production process. Those characteristics can be found, for instance, in paint processing. In paint processing plants, large batches could be technologically prohibitive. Therefore, batches of the same type of paint are usually stored in large storage vessels before they are further processed or shipped (Schultmann et al, 2006). In this case, the storage life of a new batch is affected by the storage life of the material which has already been stored in the vessel. This is particularly important for some types of paint (e.g. water-based paints) whose formulation require a limited storage time to ensure some quality specifications (e.g. to avoid drying or contamination). Thus, these materials are retained in storage only for a short period of time before they are further processed or shipped. In the former case, batches are also split in order to be processed into different paint products.

It is a common practice to formulate storage operations as tasks performed by storage vessels (see e.g. Kondili et al, 1993a; Ierapetritou and Floudas, 1998; Castro et al, 2004; Janak et al, 2004; Susarla et al, 2010). This approach also facilitates modeling multipurpose storage vessels. In general, storage tasks start upon the arrival of materials and end when they are released. Thus, whenever a storage vessel receives or releases some material, a new storage task must be initiated. If batches are assumed to be received and released as a whole, then one can easily impose storage time limitations by limiting the duration storage tasks. Nevertheless, this assumption is usually not valid as discussed in the paint processing example. Thus, since storage tasks do not relay any information regarding the remaining storage life of the contents of the storage vessels, one should consider not only the time epochs where materials are received and released, but also the flow of materials through time, to account for the combination of storage capacity and storage time limitations.

There are only a few studies in the literature which considers material flows in storage vessels (e.g. Gimenez et al, 2009). However, they do not consider storage time limitations. It must be noted that the discussion above is irrelevant for processes where batch identities are maintained through the stages (i.e. mixing and splitting of batches are not allowed). There exists some work on these types of processes addressing storage capacity and storage time limitations. Ha et al (2000) developed a formulation for flow-shop processes that incorporates both storage capacity and storage time constraints. Sundaramoorthy and Maravelias (2008) addressed storage capacity and storage time limitations in multistage processes and developed a mixed integer linear programming (MILP) model for the simultaneous batching and scheduling with storage constraints. However, it might be evident that these methods are not applicable to general multiproduct/multipurpose batch processes.

One major feature of batch scheduling models is the time representation. There are two widely used time representations: discrete and continuous time. Both time representations are based on dividing the planning horizon into a number of time intervals. In discrete time models, these intervals have fixed and equal durations, whereas in continuous time models, they have unequal durations which are not known beforehand. The choice for one of these types of modeling depends on the situation on hand as both have advantages and disadvantages. See Floudas and Lin (2004) for a general discussion on these approaches. In principle, continuous time models are more realistic and yield more precise solutions compared to discrete time models. That is because task durations are accurately accounted for in continuous time models whereas they need to be approximated in discrete time models. Furthermore, continuous models are smaller than their discrete counterparts since they require fewer number of time intervals. However, continuous time models are usually difficult to solve because their linear programming (LP) relaxations are poor and the number of time intervals is unknown (Maravelias and Grossmann, 2003). Furthermore, since the durations of time intervals are not fixed in continuous time models, intermediate due dates are difficult to model, and more importantly nonlinear expressions have to be used in order to express inventory costs, i.e. holding and backlog costs. Since these aspects are of importance in many practical cases, discrete time models remain being used in industrial problems despite their disadvantages in terms of preciseness and size (Maravelias and Grossmann, 2006). In this respect, it is valuable to stylize discrete time models to account for more realistic settings and to exploit their specific characteristics to reduce the associated computational complexities (see e.g. Shah et al, 1993; Kelly and Zyngier, 2007; Gaglioppa et al, 2008).

In this study, we propose a discrete time formulation for batch processes with storage capacity and storage time limitations. We build on the conventional state-task-network (STN) formulation of Kondili et al (1993a) which has served as a foundation for most of the work in this field.

The paper is organized as follows. In Section 3.2, we present the conventional discrete time formulation of batch processes. In Section 3.3, we highlight some aspects on the flow of materials. In Section 3.4, we formulate the storage capacity and storage time constraints for a variety of storage configurations. In Section 3.5, we present some examples to illustrate the application of the model. Finally, in Section 3.6, we provide an overall assessment of the proposed method and suggestions on directions for further research.

3.2 Formulation

One of the most important developments in modeling planning and scheduling in batch processes has been the introduction of the *state-task-network* (STN) representation of batch processes by Kondili et al (1993a). The STN approach is based on state and task nodes. The former represents the feeds, i.e. materials, and the latter represents the processing operations. The STN is defined by a number of parameters related to the tasks, states, and available equipments. Kondili et al (1993a) and subsequently Shah et al (1993) provided efficient discrete time MILP formulations for scheduling operations in STN's. In this study, we extend their formulation by incorporating storage capacity and storage time limitations. We aim at determining scheduling decisions, i.e. the start and end time of operations on specific production and storage units and corresponding production and storage quantities, minimizing the sum of production, setup, and inventory holding costs, while meeting demand under production and storage capacity limitations. In this section we neglect the storage capacity and the storage time limitations and present the basic formulation proposed by Kondili et al (1993a).

The scheduling problem in batch processes can be defined on a set of tasks $i \in I$, a set of states, $s \in S$, and a set of time periods $t \in T$. The production operations are carried out by a set of processing units $j \in J$. Each processing unit is capable of undertaking a set of tasks. The subset of processing units which are able to undertake task i are denoted by J_i. Each task produces/consumes a set of states in given proportions. We denote the set of tasks which produces/consumes state s by I_s^p/I_s^c, and the corresponding production/consumption proportions by ρ_{is}^p/ρ_{is}^c.

Each task i, when performed on processing unit j, requires a fixed processing time p_{ij}, and incurs a setup cost g_{ij} per production run and a variable production cost c_{ij} per unit of production quantity. The system incurs a holding cost h_s per unit of state s held in inventory per time period. For each task i, there are unit dependent minimum and maximum production quantities $v_{ij}^{\min}$ and $v_{ij}^{\max}$ which bound batch sizes from below and above. The demand for state s at time period t is denoted by d_{st}. Decision variables include: binary variable w_{ijt} which is equal to 1 if task i is initiated at time period t on processing unit j, and 0 otherwise, continuous variable b_{ijt} which represents the batch size of task i which is initiated at time period t on processing unit j, and continuous variable x_{st} the inventory level of state s in period t. The batch processes scheduling problem can be formulated as follows.

$$\min \quad \sum_{i\in I}\sum_{j\in J_i}\sum_{t\in T} g_{ij}w_{ijt} + \sum_{i\in I}\sum_{j\in J_i}\sum_{t\in T} c_{ij}b_{ijt} + \sum_{s\in S}\sum_{t\in T} h_s x_{st} \tag{3.1}$$

$$\text{s.t} \quad \sum_{i\in I_j}\sum_{t'=t-p_{ij}+1}^{t} w_{ijt'} \leqslant 1 \quad \forall j\in J, \forall t\in T \tag{3.2}$$

$$v_{ij}^{\min} w_{ijt} \leqslant b_{ijt} \leqslant v_{ij}^{\max} w_{ijt} \quad \forall j\in J, \forall i\in I_j, \forall t\in T \tag{3.3}$$

$$\begin{aligned} x_{st} =& x_{s(t-1)} + \sum_{i\in I_s^p}\sum_{j\in J_i} \rho_{is}^p b_{ij(t-p_{ij})} \\ & - \sum_{i\in I_s^c}\sum_{j\in J_i} \rho_{is}^c b_{ijt} - d_{st} \quad \forall s\in S, \forall t\in T \end{aligned} \tag{3.4}$$

$$w_{ijt}\in\{0,1\}; b_{ijt}, x_{st}\in\mathbb{R}_+ \quad \forall i\in I, \forall j\in J, \forall s\in S, \forall t\in T \tag{3.5}$$

The model encompasses the following sets of equations. Eq. (3.1) sets the objective function which minimizes the sum of setup, production, and holding costs. Eq. (3.2) ensures that at most one task is assigned to a unit at any time. Eq. (3.3) sets the lower and the upper bounds of batch sizes. Eq. (3.4) expresses the material balance. Eq. (3.5) sets variable domains.

3.3 Material flows in storage vessels

The formulation provided in Section 3.2 has two implicit assumptions on intermediates: (i) there is no limitation on the storage capacity of any material, and (ii) materials can be stored for unlimited time. In the following sections we extend this formulation by relaxing these assumptions. However, there are several aspects worth highlighting before analyzing the constraints regarding the storage operations.

Both storage capacity and storage time constraints affect the timing and the volume of inflow and outflow transactions in storage vessels. Storage capacity constraints limit the amount of material stored in storage vessels, whereas storage time constraints limit the amount of time that materials can be stored. In order to characterize these limitations, we assume that (i) each state s has a specific storage life of π_s periods during which it has to be sent to upstream processing stages, and (ii) when materials with different storage lives are mixed in a storage vessel the remaining storage life of fresh material degrade to the one of dated material.

An important point in modeling material flows in storage vessels is the sequence of inflow and outflow transactions. For convenience, we assume that outflow transactions take place before inflow transactions. Let us consider a storage vessel which is used to store a given state at a given period. Let `initialinv` be the inventory level at the beginning of period, and `inflow` and `outflow` be the respective volumes of the inflow and the outflow in the given period. The outflow can be satisfied either with dated material (i.e. inventory) or with fresh material (i.e. inflow). Since we consider finite wait policies, it is logical to assume that flows are coordinated by the first-in-first out principle, that is, the outflow is first satisfied by inventory, and the remaining amount (if non-zero) is satisfied by the inflow. Hence, the volume of the outflow met by inventory is $\min\{\texttt{initialinv}, \texttt{outflow}\}$. If the inventory level is higher than the outflow, then only dated materials will be used. Otherwise, after depleting all inventory, the inflow will be used to balance the outflow. Hence, the outflow met by fresh material is $\max\{\texttt{outflow} - \texttt{initialinv}, 0\}$. After inflow transactions, the inventory is replenished to $\texttt{initialinv} - \texttt{outflow} + \texttt{inflow}$. Notice that this actually represents the end-of-period inventory level (see Eq. (3.4)).

In terms of storage life limitations, the above mentioned analysis of material flow transactions has an important result. That is, when all dated material is used to fulfill the outflow (i.e. $\texttt{initialinv} \leqslant \texttt{outflow}$) fresh material can be stored without being mixed with dated material. Otherwise, fresh and dated materials are mixed in storage which results in degradation of fresh material. In the next section we will use this observation in modeling storage operations.

3.4 Modeling storage capacity and storage time constraints

The approach that must be taken to model storage time constraints strongly depends on the storage settings. In this section we consider three main cases. In the first case, each state is stored in a single dedicated storage vessel. The second case extends the first one with multiple storage vessels dedicated to each state. The third case is the most general one which addresses multipurpose storage vessels that can be used for multiple states.

3.4.1 Single storage vessel dedicated to each state

The first case addresses the instance where a single storage vessel with a given capacity is dedicated to each state. We denote the capacity of the storage vessel dedicated to state s by u_s. It is straightforward to set storage capacity constraints. That is,

$$x_{st} \leqslant u_s \quad \forall s \in S, \forall t \in T. \tag{3.6}$$

In order to integrate storage time constraints, it is essential to keep track of the remaining storage life of materials. More specifically, the periods where the entire material stock in a vessel is depleted by outflow transactions has to be specified explicitly. As we have discussed, only in such periods fresh material can be stored without being mixed with dated material. We refer to these periods as *renewal periods*. Let us denote a renewal period by $\sigma_{st} \in \{0, 1\}$, such that, $\sigma_{st} = 1$ if period t is a renewal period for state s, and $\sigma_{st} = 0$ otherwise. Then, if material flows are coordinated by first-in-first out principle, the following inequality holds.

$$x_{s(t-1)} - \sum_{i \in I_s^c} \sum_{j \in J_i} \rho_{is}^c b_{ijt} - d_{st} \leqslant u_s(1 - \sigma_{st}) \quad \forall s \in S, \forall t \in T \tag{3.7}$$

Eq. (3.7) states that, if period t is a renewal period for state s, then the inventory level after output transactions should be 0. Otherwise it is bounded by the storage capacity. Note that each period t with $x_{s(t-1)} = 0$ is a renewal period for state s.

The concept of renewal periods facilitates modeling storage time constraints. Since, the storage time limitation is defined on the number of periods that materials can be stored in a storage vessel, it can be modeled as an upper bound on the number of consecutive non-renewal periods. Hence, storage time limitation can be guaranteed in terms of the following constraint.

$$\sum_{t'=t}^{t+\pi_s-1} \sigma_{st'} \geqslant 1 \quad \forall s \in S, \forall t \in T \tag{3.8}$$

where π_s ($\pi_s \geqslant 1$) is the storage life of state s. Eq. (3.7) ensures that the remaining storage life of state s never hits 0.

3.4.2 Multiple storage vessels dedicated to each state

The second case considers the instance where multiple storage vessels with given capacities are dedicated to each state. In processes where degradation of the materials due to waiting time is negligible, materials from different batches can be considered the same regardless of the time they are processed. In such cases, multiple dedicated vessels can be modeled as a single aggregate dedicated storage vessel since mixing materials in storage does not lead to any degradation. However, in case of storage time limitations, it is possible to separate materials with different storage lives, which in turn helps to subdue the effects of storage time limitations.

Let us consider a set of storage vessels $k \in K$ each with a capacity of q_k. Which of these storage vessels k are being used for state s is known in advance. Hence, we denote the set of storage vessels dedicated to state s by K_s. Using multiple storage vessels necessitates to trace both the amount and the remaining storage life of material in each storage vessel. This can be done by defining capacity constraints, inflow/outflow transactions, and renewal periods explicitly on storage vessels rather than on states. Let us start with capacity constraints. We denote the amount of material stored in vessel k at period t by y_{kt}, such that, $\sum_{k \in K_s} y_{kt} = x_{st}$. Now, we can write storage vessel specific capacity constraints as

$$y_{kt} \leqslant q_k \quad \forall k \in K, \forall t \in T. \tag{3.9}$$

By using the new variable y_{kt} we can characterize the inflow/outflow transactions and renewal periods for each storage vessel. Let f^i_{kt}/f^o_{kt} be the volume of inflow/outflow in storage vessel k at period t. Then, while preserving the aggregate balance constraints (see Eq. (3.4)), we add a balance constraint for each storage vessel.

$$y_{kt} = y_{k(t-1)} + f^i_{kt} - f^o_{kt} \quad \forall k \in K, \forall t \in T \tag{3.10}$$

It is clear that the volumes of inflow/outflow transactions in storage vessels must match the aggregate balance. This can be guaranteed by means of the following equations.

$$\sum_{k \in K_s} f^i_{kt} = \sum_{i \in I^p_s} \sum_{j \in J_i} \rho^p_{is} b_{ij(t-p_{ij})} \quad \forall s \in S, \forall t \in T \tag{3.11}$$

$$\sum_{k \in K_s} f^o_{kt} = \sum_{i \in I^c_s} \sum_{j \in J_i} \rho^c_{is} b_{ijt} + d_{st} \quad \forall s \in S, \forall t \in T \tag{3.12}$$

In order to specify renewal periods on storage vessels, we use the binary variable σ_{kt}. Then, the following inequality ensures that $\sigma_{kt} = 1$ if the inventory level after output transactions is 0.

$$y_{k(t-1)} - f^o_{kt} \leqslant q_k(1 - \sigma_{kt}) \quad \forall k \in K, \forall t \in T \tag{3.13}$$

Eq. (3.13) is equivalent to Eq. (3.7) which was used in formulating renewal periods for a single dedicated storage vessel. The only difference is that, here renewal periods are defined on storage vessels rather than states. Subsequently, storage time limitations can be imposed on each storage vessel by means of the following constraint.

$$\sum_{t'=t}^{t+\pi_s-1} \sigma_{kt'} \geqslant 1 \quad \forall k \in K_s, \forall t \in T \tag{3.14}$$

As can be observed, the difference between modeling single and multiple dedicated storage vessels lies in additional decisions on the allocation of aggregate inflow and outflow of each state s to dedicated storage vessels, while taking the storage time and storage capacity limitations into account.

3.4.3 Multipurpose storage vessels

The third case addresses the most general instance where a number of multipurpose storage vessels can be used for any state. Multipurpose storage vessels are also considered by Kondili et al (1993a). They model storage operations as tasks which receive a certain amount of material and produce an equal amount after exactly one time period. However, with their approach, it is not possible to trace the storage life of materials in storage vessels and to address storage life limitations. For the purposes of this study, we explicitly model storage vessels. We build on the approach we have provided for modeling multiple dedicated storage vessels.

Let $k \in K$ be the set of multipurpose storage vessels, and q_k be the capacity of storage vessel k. Unlike the case with multiple dedicated storage vessels, here we do not know which storage vessels are going to be used for which state in advance. Similar to the approach taken for processing units, we introduce the binary assignment variables z_{skt}, such that, $z_{skt} = 1$ if storage vessel k is assigned to state s in period t, and $z_{skt} = 0$ otherwise. Since a storage vessel can only be assigned to one state at a time:

$$\sum_{s \in S} z_{skt} \leqslant 1 \quad \forall k \in K, \forall t \in T. \tag{3.15}$$

Consecutively, we define y_{skt} as the amount of state s stored in storage vessel k in period t, such that $\sum_{k \in K} y_{skt} = x_{st}$. Note that y_{skt} can only be non-zero if $z_{skt} = 1$. This limitation, combined with the capacity constraints, can be expressed by means of the following constraint.

$$y_{skt} \leqslant q_k z_{skt} \quad \forall s \in S, \forall k \in K, \forall t \in T \tag{3.16}$$

Similar to the multiple dedicated storage vessels case, we define the inflow/outflow transactions and renewal periods on each storage vessel. Let f^i_{skt}/f^o_{skt} be the volume of inflow/outflow of state s in storage vessel k in period t. Note that we have used three indices to characterize material flows since the state stored in a particular storage vessel is a decision variable in case of multipurpose storage vessels. Now, we can write the storage vessel specific balance constraints as follows.

$$y_{skt} = y_{sk(t-1)} + f^i_{skt} - f^o_{skt} \quad \forall s \in S, \forall k \in K, \forall t \in T \tag{3.17}$$

The storage vessel specific balance must match the aggregate balance. This can be guaranteed by means of the following equations.

$$\sum_{k \in K} f^i_{skt} = \sum_{i \in I^p_s} \sum_{j \in J_i} \rho^p_{is} b_{ij(t-p_{ij})} \quad \forall s \in S, \forall t \in T \tag{3.18}$$

$$\sum_{k \in K} f^o_{skt} = \sum_{i \in I^c_s} \sum_{j \in J_i} \rho^c_{is} b_{ijt} + d_{st} \quad \forall s \in S, \forall t \in T \tag{3.19}$$

We have used extra state indices in modeling material flows. This is not necessary to model renewal periods. It is sufficient to guarantee that the storage vessel is fully depleted after outflow transactions regardless of the state. Let us use binary variable σ_{kt} to specify renewal periods on storage vessel. Then,

$$y_{sk(t-1)} - f^o_{skt} \leqslant q_k(1 - \sigma_{kt}) \quad \forall s \in S, \forall k \in K, \forall t \in T. \tag{3.20}$$

It is not possible to impose storage time restrictions for multipurpose storage vessels in the same fashion as in the case of multiple dedicated storage vessels. This is due to the fact that storage times are state specific whereas renewal periods are not. Consequently, storage time restriction can be satisfied by imposing at least one renewal period for state s in every block of π_s periods. This can be done by means of the following inequality.

$$\sum_{t'=t}^{t+\pi_s-1} (z_{skt'} - \sigma_{kt'}) < \pi_s \quad \forall s \in S, \forall k \in K, \forall t \in T \tag{3.21}$$

Note that Eq. (3.21) is only binding when storage vessel k is assigned to state s within periods t and $t + \pi_s - 1$. If this is the case, then there must be at least one renewal period for state s within this interval.

3.5 Computational experiments

In this section we aim to illustrate how storage capacity and storage time limitations affect production and storage operations and corresponding cost figures by means of numerical experiments. The instances we use are adopted from the literature (i.e. Sahinidis and Grossmann, 1991b) and stylized for the purposes of this study. In order to analyze the sole effects of storage limitations on processing and storage operations, we assume that sufficient amount of raw material is available whenever needed (i.e. we do not consider raw material inventories), and production costs (i.e. processing and setup costs) are not unit dependent. In this setting, there is no need to consider processing costs, and the total cost function is then composed of the sum of setup and holding costs. This enables us to analyze the effects of storage capacity and storage time limitations by considering the trade-off between setup and holding costs.

In what follows, we provide three illustrative numerical experiments. In each experiment, we demonstrate a different aspect of storage limitations by using several scenarios with different storage configurations. In Experiment A, we analyze the effects of storage limitations on the number of setups and average inventories by considering the case where a single storage vessel is dedicated to each state. In Experiment B, we show that the storage time limitations can be very restrictive due to the degradation of materials particularly when there is a single dedicated storage vessel for each state. Then, we also show how the use of multiple storage vessels can help to reduce the effects of these restrictions. In Experiment C, we analyze the efficiency of using multipurpose storage vessels in comparison with using dedicated storage vessels.

3.5.1 Experiment A – single storage vessel dedicated to each state

In this experiment, we illustrate how storage limitations affect production and storage operations. We consider the problem instance defined by the STN given in Figure 3.1 and the specifications provided in Table 3.1. The system involves a single intermediate and two end products. There are three processing units each of which can be used to perform one of the three tasks. We consider four scenarios with different storage configurations: unlimited storage capacity and unlimited storage time (Scenario-A1), limited storage capacity and unlimited storage time (Scenario-A2), unlimited storage capacity and limited storage time (Scenario-A3), and limited storage capacity and limited storage time (Scenario-A4). In each of those scenarios, there are three storage vessels each of which is dedicated to one of the three states. The storage configurations of these scenarios are given in Table 3.2. We analyze the effects of storage capacity and storage time limitations on the number of setups for each task and the average level of inventories for each state. Table 3.3 presents the results for each scenario. The results clearly shows that both storage capacity and storage time limitations have significant effects on the resulting optimal schedule and the corresponding cost figure.

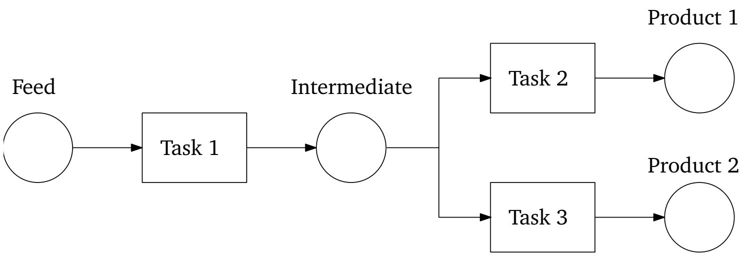

Figure 3.1: State-task-network of Experiment A

Scenario-A1 expresses the case with no restrictions on storage, and hence, its optimal cost 1605 constitutes a lower bound for the other scenarios. The storage capacity and storage time limitations progressively increase this cost figure. Introducing storage capacity constraints in Scenario-A2 leads to one extra setup for each of the three tasks and increases the total cost to 1962. Introducing storage time constraints in Scenario-A3 results in one extra setup for Task 1 and Task 2, and two extra setups for Task 3, and increases the total cost to 2162. Introducing both storage capacity and storage time constraints in Scenario-A4 adds two extra setups for Task 1 and Task 3, and one extra setup for Task 2, and increases the total cost

Processing Units

	Min. Batch Size	Max. Batch Size	Suitability	Proc. Time
Unit 1	0	1500	Task 1	1
Unit 2	0	1000	Task 2	1
Unit 3	0	1000	Task 3	1

Demands

	1	2	3	4	5	6	7	8	9	10	11	12
Product 1				300			450			600	150	
Product 2				75		225				300		150

Costs

Setup cost	200
Holding cost	0.18

Table 3.1: Data used in Experiment A

	Storage capacity			Storage time (periods)		
	Interm.	Prod. 1	Prod. 2	Interm.	Prod. 1	Prod. 2
Scenario-A1	∞	∞	∞	∞	∞	∞
Scenario-A2	200	400	150	∞	∞	∞
Scenario-A3	∞	∞	∞	1	1	1
Scenario-A4	200	400	150	1	1	1

Table 3.2: Storage configurations of the four scenarios of Experiment A

to 2281. In all scenarios, introducing storage capacity and storage time limitations tend to reduce the average inventory levels. To summarize, we observe that storage constraints result in a reduction of inventories by initiating more frequent setups, and they significantly degrade the cost performance of the system.

	# of setups			Average inventory			Total cost
	Task 1	Task 2	Task 3	Interm.	Prod. 1	Prod. 2	
Scenario-A1	2	2	2	0.00	125.00	62.50	1605
Scenario-A2	3	3	3	33.33	12.50	29.16	1962
Scenario-A3	3	3	4	31.25	12.50	31.25	2162
Scenario-A4	4	3	4	12.50	12.50	12.50	2281

Table 3.3: Results of the four scenarios of Experiment A

3.5.2 Experiment B – multiple storage vessels dedicated to each state

In this experiment, we demonstrate the criticality of storage time limitations. We consider the problem instance defined by the STN given in Figure 3.2 and the parameters provided in Table 3.4. This system involves three tasks each of which processes a certain feed into an end product. Since we assume that feeds are available whenever they are needed, we only analyze the production and storage operations of three end products. There are two processing units: Unit 1 which is suitable for Task 1 and Task 3, and Unit 2 which is suitable for Task 2 and Task 3. We assume neither storage capacity nor storage time limitations for Product 1 and Product 2, and we concentrate on Product 3 inventories. We consider three scenarios with different storage configurations for Product 3: unlimited storage capacity and storage time (Scenario-B1), a single dedicated vessel (with unlimited capacity) and limited storage time (Scenario-B2), and two dedicated storage vessels (with limited capacities) and limited storage time (Scenario-B3). The details of the storage configurations are given in Table 3.5. This experiment mainly focuses on the comparison between Scenario-B2 and Scenario-B3. The results show that using two smaller storage vessels rather than a very large one could be more efficient in presence of the storage time limitations.

The system sketched in Experiment B leads to a very tight production schedule. In fact, the solution pool of Experiment B includes only a single feasible solution in Scenario-B1 which reflects the case with no limitation in storage. Therefore,

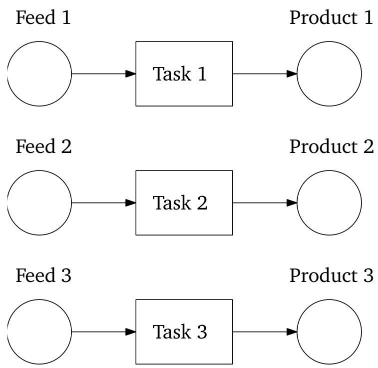

Figure 3.2: State-task-network of Experiment B

the following two scenarios are either infeasible, or they are optimal and use the same production schedule as in Scenario-B1. This production schedule is depicted in Figure 3.3. Let us consider Product 3 inventories through the planning horizon. As can be seen in Figure 3.3, Product 3 has to be released in two batches of Task 3. The first batch has a size of 250 and it has to be released in Period 3. The second batch has a size of 100 and it has to be released in Period 5. The first batch is initially used to fulfill the demand of size 100 in Period 3. The remaining amount of this batch together with the second batch is then used to fulfill the demands in Period 6 and 9 with the respective sizes of 200 and 50. The resulting inventory levels of Product 3 are illustrated in Figure 3.4.

In Scenario-B2 we introduce the storage time limitation of 4 time periods and assume that Product 3 has a single dedicated storage vessel with infinite capacity. However, this scenario has no feasible solution since the only feasible schedule involves a block of 5 consecutive periods without a renewal period for Product 3. This is due to the fact that the two batches of Product 3 have to be mixed in a storage vessel since there is only a single storage vessel available for Product 3.

In Scenario-B3 we replace the single dedicated vessel with two dedicated vessels with the respective capacities 100 and 150. This modification makes it possible to separate the two batches of Product 3, and consequently, leads to a feasible schedule. This is illustrated in Figure 3.5. This experiment clearly presents a case where using two smaller storage vessels can be more efficient than using a single storage vessel with a very large volume because of the storage time limitations.

Processing Units

	Min. Batch Size	Max. Batch Size	Suitability	Proc. Time
Unit 1	0	250	Task 1, Task 3	6, 2
Unit 2	0	100	Task 2, Task 3	2, 2

Demands

	1	2	3	4	5	6	7	8	9
Prod. 1									250
Prod. 2			100				100		100
Prod. 3			100			200			50

Costs

Setup cost	100
Holding cost	0.1

Table 3.4: Data used in Experiment B

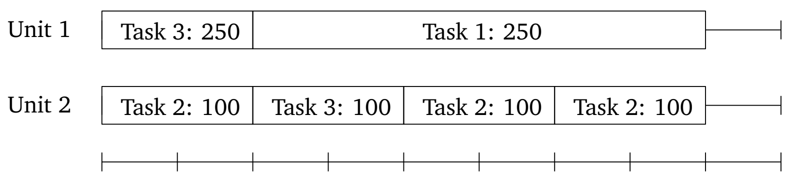

Figure 3.3: The feasible production schedule in Scenario-B1

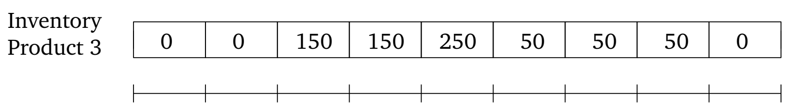

Figure 3.4: The inventory levels of Product 3 in Scenario-B1

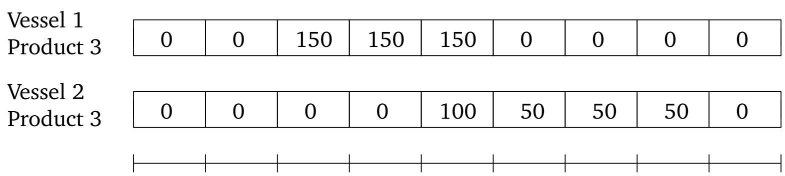

Figure 3.5: The feasible storage schedule of Product 3 in Scenario-B3

	Product 3	
	Storage vessels (#: capacity)	Storage time (periods)
Scenario-B1	1:{Unlimited}	Unlimited
Scenario-B2	1:{Unlimited}	4
Scenario-B3	2:{150,100}	4

Table 3.5: Storage configurations of the three scenarios of Experiment B

3.5.3 Experiment C – multipurpose storage vessels

In this experiment, we compare the respective cases with dedicated and multipurpose storage vessels and analyze their efficiency in terms of operational costs. We consider the problem instance defined by the STN given in Figure 3.6 and the parameters provided in Table 3.6. The system involves eight tasks, six intermediates and four end products. There are six processing units which are available to carry out the set of tasks. In order to highlight the efficiency of multipurpose storage vessels on cost performance, we consider three scenarios with different storage settings. In each of those scenarios, the maximum storage time of all states are set to three periods. The scenarios include the following storage settings: ten dedicated storage vessels – one for each state (Scenario-C1), five multipurpose storage vessels (Scenario-C2), and four multipurpose storage vessels (Scenario-C3). Both dedicated and multipurpose vessels have the capacity of 500. Table 3.7 presents the results for each scenario including the total number of setups, the total average level of inventories, and the total costs. The results reveal that multipurpose storage vessels are dramatically more efficient than dedicated storage vessels in terms of cost performance.

In Scenario-C1 we sketch a rigid storage setting with a rather large capacity (10 vessels each with capacity 500). The optimal schedule of Scenario-C1 involves 12 setups and an average inventory level of 688.80 which leads to the cost of 2951. Scenario-C2 illustrates an alternative storage setting characterized by flexible storage vessels however with half of the capacity considered in Scenario-C1 (5 vessels each with capacity 500). The optimal schedule of Scenario-C2 involves 11 setups and an average inventory of 670.94 which results in a cost of 2736. By comparing the first two scenarios, we observe that even five multipurpose storage vessels outperform ten dedicated storage vessels the in terms of cost efficiency. In Scenario-C3 we decrease the number of multipurpose storage tanks from five to four. This necessitates three extra setups while reducing the average inventory level to 594.88 in

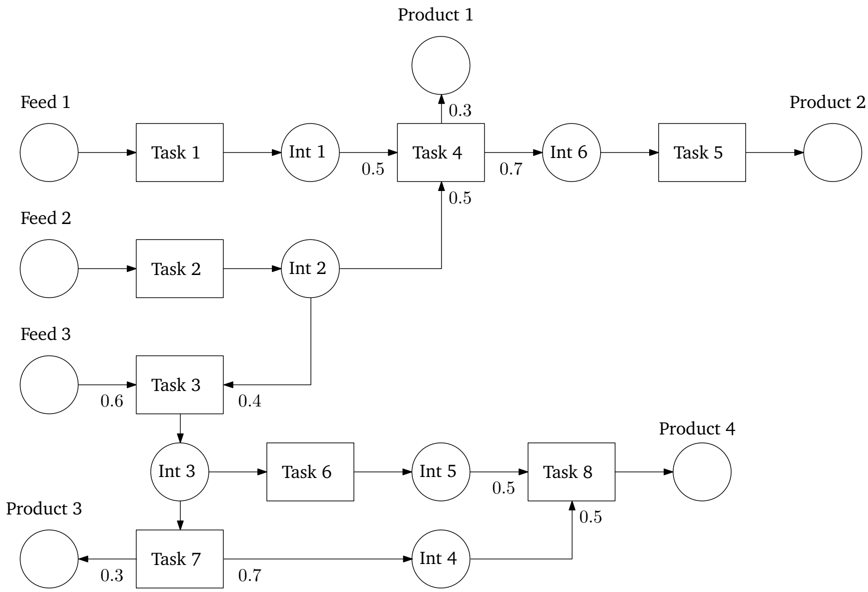

Figure 3.6: State-task-network of Experiment C

Processing Units

	Min. Batch Size	Max. Batch Size	Suitability	Proc. Time
Unit 1	0	1000	Task 1	1
Unit 2	0	2500	Task 3,7	1
Unit 3	0	3500	Task 4	1
Unit 4	0	1500	Task 2	1
Unit 5	0	1000	Task 6	1
Unit 6	0	4000	Task 5,8	1

Demands

	1	2	3	4	5	6	7	8
Product 1			110	110	133.3	100	33.3	33.3
Product 2				233.1	260		360	360
Product 3					116.6	56.6		116.6
Product 4						333.3	333.3	685.8

Costs

Setup cost	200
Holding cost	0.1

Table 3.6: Data used in Experiment C

comparison with Scenario-C2. Hence, Scenario-C3 results in a cost of 3276 which is larger than the cost in Scenario-C1. These results illustrate that using multipurpose storage vessels could be dramatically more efficient than using dedicated storage vessels. For this particular experiment, we observe that the cost performance of using dedicated storage vessels can be achieved by using multipurpose storage vessels with half of the storage capacity.

	Storage configuration	Total # of setups	Total average inventory	Total cost
Scenario-C1	10 × 500 (dedicated)	12	688.80	2951
Scenario-C2	5 × 500 (multipurpose)	11	670.94	2736
Scenario-C3	4 × 500 (multipurpose)	14	594.88	3276

Table 3.7: Results of the three scenarios of Experiment C

It should be noted that there are many other factors of practical interest which are not considered in the current numerical study. For instance, in many production environments storage vessels require setup times and costs. In such cases, using a large number of storage vessels with rather limited capacities may degrade the overall performance due to extensive setup operations. Thus, the results reported here can be broadened to better understand the effects of further factors on the system performance.

3.5.4 Computational statistics

All problem instances are solved using CPLEX 11.1 in OPL Studio 6.0 modeling environment on a 1.83 GHz computer with 1.00 GB of RAM. Model and solution statistics for all instances are given in Table 3.8.

3.6 Conclusions and extensions

In this paper we proposed a discrete time formulation for scheduling multiproduct/multipurpose batch processes considering storage capacity and storage time limitations which are very common in many industries involving perishable intermediates and end products. We explicitly modeled storage vessels on which we defined material flows and expressed storage constraints. We have formulated the problem mathematically as a MILP model building on the conventional formulation

Instance	Variables		Constraints	Nodes	CPU (s)
	Binary	Continuous			
Scenario-A1	108	147	399	0	0.25
Scenario-A2	108	147	435	0	0.25
Scenario-A3	144	147	471	0	0.25
Scenario-A4	144	147	507	0	0.25
Scenario-B1	54	84	210	0	0.00
Scenario-B2	81	84	264	0	0.00
Scenario-B3	90	196	439	0	0.00
Scenario-C1	464	474	1450	26	0.25
Scenario-C2	824	1714	3170	3564	65.00
Scenario-C3	736	1464	2842	14417	337.76

Table 3.8: Solution statistics of all instances

of Kondili et al (1993a), and showed how the proposed approach can account for specific storage configurations such as single/multiple dedicated and multipurpose storage vessels. By means of numerical experiments, we illustrated how storage capacity and storage time limitations affect production and storage operations. We showed that storage limitations can significantly reduce the cost performance of processing systems. We also demonstrated that the usage of multipurpose storage vessels can significantly help to overcome storage limitations.

There has been a significant amount of work done in the literature on exploiting various specific features of discrete time scheduling problems. Since the proposed model inherits the underlying features of general discrete time scheduling models, it is easy to extend it with the approaches which have already been proposed in the literature. We shortly discuss some of them. For both production and storage units, sequence-dependent and/or frequency-dependent setups can be incorporated into our formulation following to Kondili et al (1993a) or Kelly and Zyngier (2007). Makespan minimization can be set as an objective function by following the approach of Maravelias and Grossmann (2003). In order to reduce the computational time, reformulations and valid inequalities proposed by Shah et al (1993), and recently by Gaglioppa et al (2008) can be applied.

There are several directions for further research worth exploring. First, the framework developed in this study can be applied to continuous time models. One major difficulty here is formulating material flows in storage vessels. Only recently, Gimenez et al (2009) developed a novel continuous time formulation which also considers the material flows in storage vessels. The approach we used in this study could also be applied in their continuous time framework. Secondly, the approach taken in this study can be extended to continuous or semi-continuous processes. The literature suggests well-established approaches for scheduling these processes which could possibly be extended to address the combination of storage capacity and storage time limitations. Thirdly, a different modeling paradigm e.g. constraint programming (CP) can be employed together with MILP in order to solve the mathematical problem more efficiently. In the last decade successful applications of MILP/CP hybrid approaches have been reported for similar problems (see e.g. Jain and Grossmann, 2001).

Chapter 4

Scheduling a two-stage evaporated milk production process

Abstract. *This paper addresses a process scheduling problem originating from a processing system specialized in evaporated milk products. The layout of the system involves two continuous processing stages: processing and packaging. The processed materials are batch-wise standardized in between these production stages. The production environment has several industry-specific characteristics involving traceability requirements and time- and sequence-dependent cleaning of production units. These all together lead to a challenging scheduling problem which requires an efficient and yet flexible modeling approach. A two-phase mathematical approach is presented for this problem which successively determines the specifications regarding material flows and builds a complete production schedule. The approach is then shown to be efficient by means of a numerical study based on the data collected from a real-life evaporated milk plant.*

4.1 Introduction

The process industry has evolved towards more market oriented strategies in the last decades in response to the trends with respect to larger product variety and intense competition. This resulted in a growing interest in processing systems which provides the flexibility of handling demands with regard to a variety of products while using limited production resources. Production scheduling in flexible processing systems is more demanding as compared to dedicated flow type processing systems. Hence, the impact of effective scheduling methods is significant. Process scheduling problems have received considerable attention in the literature. The vast majority of research contributions have been published in the chemical engineering literature and centered on scheduling of batch processes in large-scale chemical plants. A comprehensive overview of state-of-the-art models and methods in this line of research can be found in Kallrath (2002), Floudas and Lin (2004), and Mendez et al (2006).

In this paper, we investigate a process scheduling problem originating from a processing plant of a dairy company specialized in evaporated milk products. The production layout of the system involves two continuous production stages where fresh milk is first processed into evaporated milk and then put into consumer packaging. These production stages are connected by storage tanks where materials are batch-wise standardized. The production system involves the processing of a large variety of product recipes based on the main raw material fresh milk. The product recipes mainly differ in terms of their dry-matter concentration. The number of end-products is much larger than the number of production recipes due to different packaging types.

The production process under consideration has some distinctive characteristics that require careful consideration in the context of scheduling. The production resources associated with both the processing and the packaging stages as well as the intermediate storage are strictly limited. The production resources require sequence- as well as time-dependent cleaning due to varying dry-matter concentrations of different product recipes and hygiene requirements. The traceability of materials through the production process is an important concern. The company policy towards traceability calls for the integrity of materials used in each customer order. These, all together lead to a highly constrained operational environment that, to the best of our knowledge, cannot be captured by readily available scheduling approaches in the literature. The literature has paid a great deal of attention to some of the aforementioned characteristics. For example, already in earlier contributions on process scheduling sequence-dependent cleaning is considered as one of the basic characteristics of processing systems (see e.g. Kallrath, 2002, and references therein). There are even studies which explicitly address the issue of modeling sequence dependency (see e.g. Kelly and Zyngier, 2007). However, the research efforts towards some other characteristics are rather limited. There are only a few papers which address time-dependent cleaning (see e.g. Kondili et al, 1993a; Papageorgiou and Pantelides, 1996a). The literature on operations management and operations research has only recently started to take over the concept of traceability. There are only a few studies which reflect the effects of traceability on the operational performance (see e.g. Wang et al, 2009, 2010; Rong and Grunow, 2010). However, none of these research efforts considers traceability in the context of process scheduling with capacitated resources.

Traceability is a critical concern in the food processing industry since it contributes to food safety by facilitating product recalls when a problem is identified (Dupuy et al, 2005). It necessitates a perpetual control of the flow of materials through the production system. In the literature on process scheduling, the so-called material balance equations appear to be the standard approach towards modeling material flows (Mendez et al, 2006). These equations dictate the flow conservation, and control the accumulation and consumption of materials through time. Thus, they guarantee that materials are available when they are necessary. However, this approach does not leave any room to coordinate the execution of production operations and to keep track of the origin of materials to associate them with particular customer orders. It should be obvious that having control over those is essential to keep up with traceability requirements. Therefore, it is necessary to employ a different modeling approach towards material flows in order to account for traceability.

The purpose of this study is to present a scheduling approach which can systematically characterize and accommodate the industry-specific characteristics of the evaporated milk production process. The main feature of the approach is to decompose the problem in such a way that enables us to make use of the specific characteristics of the system in modeling and solving the integrated sub-problems. We decompose and solve the overall scheduling problem in two phases. In the first phase, we specify the number and size of the standardization batches and how those batches are used to fulfill particular customer orders. The separation of these decisions from the rest of the process scheduling problem enables us to coordinate material flows while respecting traceability requirements. In the second stage, we assign production operations to suitable production resources and determine the timing of production operations to realize the material flows specified in the first sub-problem. This integrated method leads to a simple yet flexible approach for scheduling evaporated milk production processes. The most important advantage of this approach is its simplicity. The enclosed mathematical models can easily be written and plugged into commercial solvers.

The rest of this paper is organized as follows. In Section 4.2, we give an overview of the relevant literature. In Section 4.3, we describe the production process in detail. In Section 4.4, we introduce the modeling approach. In Section 4.5, we provide a formal definition of the problem. In Section 4.7, we illustrate the applicability of the proposed formulation by means of a real-life case. Finally, in Section 4.8, we conclude and provide some directions for further research.

4.2 Relevant literature

The process layout addressed in this study falls into the category of the so-called make-and-pack systems which could be considered as a specific case of semi-continuous production processes (Mendez and Cerda, 2002; Kopanos et al, 2010). These production environments are common especially in the food processing industry, and they are often characterized by perishable materials, divergent product structures, limited production resources, and sequence-dependent cleaning requirements (Van Donk, 2001). The literature on scheduling make-and-pack production systems is rather limited. However, their practical relevance has motivated some researchers towards case oriented process scheduling problems.

Jain and Grossmann (2000) addressed a two-stage continuous process motivated by a manufacturing plant of a fast moving consumer goods company. They initially modeled the scheduling problem as a disjunctive program with the objective to minimize makespan using a continuous time representation. Then, they transformed this into a mixed integer linear programming (MILP) model. In order to reduce the computational time they introduced a set of partial pre-ordering rules. This enabled the authors to handle larger problem instances while not sacrificing optimality too much. However, their formulation is based on the assumption that each customer order is processed, stored, and packaged individually. Thus, they do not look into the case where multiple customer orders are fed by the same processing run. Furthermore, they do not consider sequence- and time-dependent changeovers.

Mendez and Cerda (2002) considered the production process of a candy manufacturing plant. They provided a continuous time MILP formulation of the associated process scheduling problem with the objective to minimize makespan. They respected various real-life specifications such as order due dates and sequence-dependent changeovers. They introduced a set of pre-ordering rules into the MILP formulation which reduce computational times while providing good feasible schedules. However, their formulation strongly relies on the assumption of unlimited storage resources between processing and packaging stages which is not the case in most production environments.

Entrup et al (2005) studied the case of yogurt production. They employed a block planning approach where a schedule is defined on a repeated cycle of a pre-defined sequence of production operations. They presented different MILP formulations optimizing a cost-based objective function using a hybrid discrete- and continuous-time representation. The formulations are tested on a case study and it is shown that near-optimal solutions can be obtained within reasonable computational times. However, their problem only concerns the scheduling of the packaging stage. Thus, operations involving the processing and storage of products are neglected.

Marinelli et al (2007) addressed the scheduling problem in a yogurt production system. They developed a discrete time capacitated lot sizing and scheduling formulation of the problem with a cost-based objective function. They proposed a heuristic based on the decomposition of the overall problem into a lot sizing and a scheduling problem which are solved successively. It is shown that the heuristic exhibits near-optimal solutions in a short computational time. However, the proposed formulation is based on the assumption that the production rate is fixed by a single bottleneck stage. Furthermore, they assumed that the changeovers are independent from the production sequence. These all together make their approach more suitable for planning rather than detailed scheduling problems.

Doganis and Sarimveis (2008) studied a yogurt production plant. They formulate the problem as a MILP model with a cost-based objective function using a hybrid discrete- and continuous-time representation. They consider various real-life limitations involving sequence restrictions and sequence-dependent changeovers. They tested and verified the efficiency of the formulation by means of a case study. However, similar to the approach of Marinelli et al (2007), they only consider the packaging stage and ignore all potential limitations regarding the rest of the processing system.

Kopanos et al (2010) addressed the scheduling problem in a yogurt production line. They developed a MILP formulation of the scheduling problem. Their formulation is essentially similar to the one of Doganis and Sarimveis (2008). They employed a cost-based objective function and used a hybrid discrete- and continuous-time representation. They implemented their approach in a case study and illustrated its efficiency. However, they impose timing and capacity constraints on the processing stage on an aggregate level, and the scheduling problem they consider only involves the packaging stage.

The brief overview of the relevant literature reveals that most research efforts towards process scheduling in make-and-pack plants concentrate on the packaging stage, and do not integrate the rest of the production system into the scheduling problem. We also observe that capacity limitations in the intermediate storage are often neglected. These could mainly be attributed to the computational complexity of the mathematical problems which reflects upon real-life processing systems as a whole. These problems can be modeled as large-scale mathematical programs. However, the respective formulations often do not lead to high quality solutions in reasonable computational times (Mendez et al, 2006). Nevertheless, the applicability of the models which focus on a single production stage strongly depends on the availability and the efficiency of the production resources carrying out the rest of the processing and/or storage operations. Another important observation derived from the literature review is that the literature does not suggest an approach towards addressing traceability in the context of process scheduling in make-and-pack plants.

4.3 Description of the production system

In this section, we provide a detailed description of the production process of evaporated milk, and discuss the specifications of the production system under consideration.

4.3.1 Production process of evaporated milk

Evaporated milk is a shelf-stable milk preserve with considerably reduced water content. It is one of the most widely available milk preserves since its nutritive content is not significantly different from fresh milk when deluded with water. It is also very attractive for transportation purposes because it is produced by reducing the water content of fresh milk and offers a much longer shelf-life.

There are several variants of evaporated milk production systems. The reader is referred to Walstra et al (2006, Chapter 19) for a detailed description of those variants. Here, we are rather interested in the production process and the production layout of the dairy plant which motivated this study. Figure 4.1 illustrates the steps within this process which we briefly discuss below.

Pre-heating. The fresh milk is heated before it is further processed. This treatment increases the stability of the milk, inactivate enzymes, and decreases the level of bacteria. It also prevents coagulation during storage. The preferred heat treatment

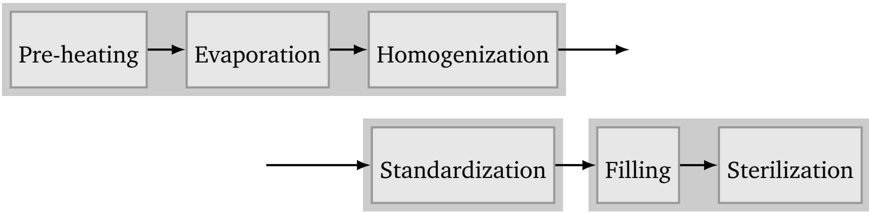

Figure 4.1: Production process of evaporated milk

is called the High Temperature Short Time (HTST) method which continuously heats the milk to temperatures exceeding 75°C for about 15 seconds.

Evaporation. The milk is concentrated by evaporation. The main concern in evaporation is to standardize the dry-matter content of the milk. Through evaporation the boiling point of the milk is significantly lowered. The warm milk continuously goes through an evaporator where it is concentrated to a lower portion of its dry-matter content.

Homogenization. The evaporated milk is homogenized by breaking large fat globules to smaller ones by forcing the milk through small holes under high pressure. This prevents creaming and coalescence while improving the color and texture of the product. The evaporated milk is continuously piped through a homogenizer immediately after evaporation.

Standardization. It is not uncommon that the viscosity of the evaporated milk does not comply with quality specifications. Thus, in order to ensure quality, a series of tests are performed on samples, and if necessary milk is standardized by using a stabilizing salt such as sodium hydrogen phosphate or potassium phosphate. This treatment is applied batch-wise. Thus further production operations need to be postponed until it is completed. The evaporated milk is kept in storage tanks during standardization. Nevertheless, long storage times are prohibitive due to the risk of bacterial growth and age thickening.

Filling. Condensed milk products are usually meant for use in regions where milk is scarce. Thus, cans are the commonly preferred packaging material due to storage and shelf life concerns. The evaporated milk is continuously piped into pre-sterilized cans which are then vacuum-sealed.

Sterilization. The evaporated milk is in-can sterilized. This treatment is mainly intended for the same purpose as pre-heating. However, it is rather extensive. Here, the milk is continuously heated to 121°C for about 8 minutes. In order to assure that the evaporated milk is heated in a standard way, cans are agitated during sterilization.

4.3.2 Specifications of the production system

The production process of evaporated milk can be encapsulated in two continuous production stages: processing fresh milk into evaporated milk and putting evaporated milk into customer packaging. In between those, evaporated milk is stored and batch-wise standardized in the intermediate storage. These primary production stages are highlighted in Figure 4.1. The layout of the processing system is organized in line with this two-stage configuration. The fresh milk first goes through processing lines where pre-heating, homogenization and evaporation take place, then standardized in storage tanks, and finally packaged and sterilized in packaging lines. This process structure is a typical example of the make-and-pack systems.

The processing plant manufactures a large variety of end products characterized by their product recipes and packaging types. In the processing stage, different types of product recipes are processed in several parallel processing lines. The processing rate varies among product recipes. Throughout processing runs, evaporated milk products are continuously piped into intermediate storage tanks where they are temporarily stored. The storage tanks are identical in size and they are capable to store products belonging to all types of product recipes. Before they are put into customer packaging, evaporated milk products are batch-wise standardized in storage tanks. The standardization of materials requires a significant amount of time. The time required to standardize a batch differs from one recipe to another but it is independent from the batch size. In the packaging stage, evaporated milk products are packaged in cans of different sizes following customer order preferences. Each standardization batch serves a single or multiple customer orders. This production stage is carried out by several parallel packaging lines each of which is dedicated to a particular can size. The packaging rates are often higher for cans with larger sizes. While they are being packaged, evaporated milk products are continuously depleted from storage tanks. It is an important concern to maintain the traceability of materials through the production process. The company policy towards traceability has important consequences on the coordination of the flow of materials: first, the standardization batches should be processed and customer orders should be packaged in uninterrupted continuous production runs, and second, each customer order should be filled by materials associated to a single standardization batch. These together lead to a divergent material flow structure which guarantees the integrity of materials used in customer orders, and ensures that materials can easily be traced back and recalled once a problem is identified.

Because of varying dry-matter concentrations of different product recipes and hygiene requirements, processing and packaging lines need to be cleaned. There are two types of cleaning. The first is the time-dependent cleaning which calls for a maximum amount of time that a production line can be used without cleaning. The second is the sequence-dependent cleaning which necessitates the cleaning of production lines whenever switching from a higher to a lower concentrated recipe. The cleaning requirements towards storage tanks are rather extensive. It is necessary to clean storage tanks after each time they are used even if the same product recipe is stored.

The dairy plant operates in a make-to-order setting because of the variety of end products and fluctuating demands. Since advance demand information is rather limited, scheduling is done on a weekly basis. The production system is mostly automated and it runs 24 hours a day and 7 days a week. Therefore, the scheduling problem of the plant involves the coordination of production operations so as to fill customer orders within a continuous time-frame of a week. The plant scheduler weekly develops a production schedule concerning the specifications and the timing of production operations and the assignment of those operations to suitable resources. The synchronization of production stages is difficult due to the difference between processing and packaging rates, and the limitations on the intermediate storage. Furthermore, the technical constraints such as cleaning and traceability requirements interfere with the timing and assignment decisions. In particular, improper scheduling decisions deprive materials from availability of suitable processing and packaging lines, and also give rise to an excess usage of storage tanks. These all together make the scheduling of the processing system a challenging task.

There is a variety of performance measures that can be adopted to assess to the quality of the production schedule. For example, it is important to limit the number of standardization batches because of the long waiting times required to guarantee the conformity of evaporated milk products. It is also an important concern to avoid extensive down times due to cleaning. However, the main managerial concern of the dairy plant is to increase the throughput of the production system. This goal is coupled with the objective to minimize the makespan which is in line with the other performance measures.

4.4 Modeling approach

We use a decomposition method to tackle the process scheduling problem under consideration. The most essential decision in designing a decomposition approach is how to break the overall problem in such a way that the resulting sub-problems can be managed efficiently while obtaining optimal or near-optimal solutions. Here, respecting the manual scheduling process currently in practice, we decompose and solve the problem in two phases: the matching phase and the scheduling phase. In these respective phases, we first determine the specifications regarding material flows, and then we build a complete production schedule to realize these material flows. The separation of these decisions enables us to coordinate material flows while respecting traceability requirements. This decomposition scheme is essentially similar to the one proposed by Neumann et al (2002) for batch process scheduling problems. However, for the purposes of this study, in the first phase we not only convert aggregate requirements into production batches, but also assign those batches to individual customer orders.

The flow of materials can be expressed in terms of two key entities: standardization batches and customer orders. Thus, the specifications (i.e. the recipe and the size) of these entities define the flow of materials. The specifications of customer orders are known in advance. The specifications of standardization batches, on the other hand, are decided upon in the matching phase. More specifically, the matching phase determines the set of standardization batches of each product recipe, and allocates those batches to particular customer orders. The standardization of an evaporated milk batch requires an extensive amount of time. Thus, it is favorable to standardize materials in large batches. Following this, in the matching phase, we aim at finding the minimum number of standardization batches that can fill all customer orders, and formulate the respective decision problem as a straightforward MILP model.

When the specifications of standardization batches and customer orders become known, it is possible to identify all types of production operations that will be carried out through the planning horizon. We distinguish three types of production operations: processing tasks, packaging tasks, and storage tasks. A processing task is a continuous processing run of a particular standardization batch, and it is carried out by a single processing line which feeds a single storage tank. A packaging task is a continuous packaging run of a particular customer order, and it is carried out by a single packaging line which is fed by a single storage tank. A storage task refers to the process of accumulation, storage, and the depletion of materials associated with a particular standardization batch, and it is carried out by a single storage tank.

Figure 4.2 illustrates the timing interactions among those operations my means of an example where a standardization batch serves three customer orders. The time axis is divided into three time periods. The first one covers the processing task. The second one is the time period between the end of the processing task and the start of the packaging task that starts the earliest. Obviously, this time period is longer than the time required to standardize the batch. The third one involves the packaging tasks. It starts with the packaging task that starts the earliest and ends with the one that ends the latest. Throughout these three time periods a storage task must be active to handle material flows.

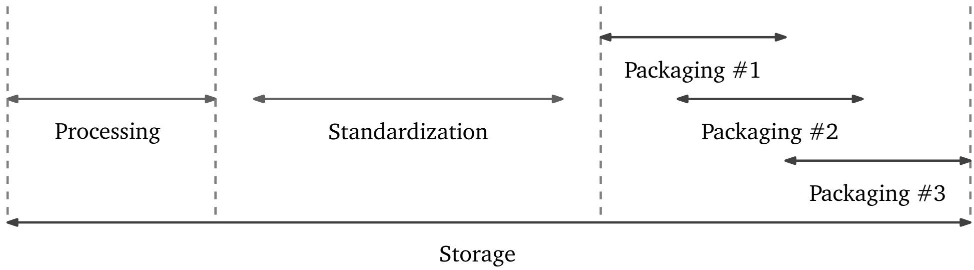

Figure 4.2: An illustration of the timeline of production operations

The scheduling phase of the proposed approach is devoted to assign production operations to suitable production resources and to determine the timing of production operations to realize the material flows specified in the matching phase. An important consideration in formulating the scheduling problem is time- and sequence-dependent cleaning requirements in production lines. We employ the block planning approach which was also used by Entrup et al (2005) to facilitate modeling those in both processing and packaging stages. The approach relies on the idea of organizing the production schedule as a repeated cycle of blocks which are associated with particular product recipes. The blocks within each cycle follow a predefined sequence which requires only a little or no cleaning effort. Thus a complete cleaning has to take place only between each cycle. Figure 4.3, illustrates the block planning approach on a fragment of the schedule of an arbitrary production line. Notice that product recipes are sequenced in increasing order on the basis of their dry-matter concentration (i.e. low-, medium-, and high-concentrated product recipes).

Entrup et al (2005) used the block planning approach in a daily framework where each cycle is coupled with a day in the planning horizon. This is a reasonable approach for systems which need to be switched on and off on a daily basis. However,

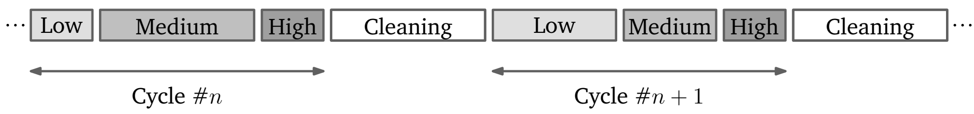

Figure 4.3: Block planning approach

the production system under consideration essentially works without interruptions. Thus, we do not pre-specify the timing of production cycles, and determine those in the context of the scheduling problem. Furthermore, we extend the block planning approach by considering time-dependent cleaning requirements. We model these by imposing upper bounds on cycle lengths. The selection of the number of cycles is also critical. There must be sufficient number of cycles to sustain flexibility in synchronizing production tasks. However, a large number of cycles should not be preferred in order to avoid extensive cleaning times. Therefore, in our formulation, we enforce an upper bound on the number of cycles in the processing and the packaging stages.

We formulate the scheduling problem by using the constraint-based modeling paradigm which is referred to as constraint programming (CP). The most widely used method in formulating process scheduling problems is MILP due to its flexibility in terms of modeling (Floudas and Lin, 2005). However, it is usually often difficult to solve large-scale real-life scheduling problems by using this exact approach. For instance, as we mentioned in the literature review, Jain and Grossmann (2000) developed a MILP formulation for a process scheduling problem similar to the one addressed in this study. Their problem is less demanding because it reflects a case where each customer order is processed and packaged as a single entity, and it does not involve time- and sequence-dependent cleaning requirements. Nevertheless, they report that finding the optimal solution for problems with more than 15 customer orders was not possible, and finding even a feasible solution for problems with more than 20 customer orders required an extensive computational time. The evaporated milk plant which has motivated this study usually receives 40 – 60 customer orders every week. This benchmark motivated us to employ a CP approach. CP is known to be capable of finding good solutions for highly constrained real-life scheduling problems within reasonable computational times (Baptiste et al, 2001). This mainly stems from the fact that it searches for feasibility rather than optimality. When used in an optimization concept, CP sequentially finds better feasible solutions by bounding the objective function.

4.5 Problem statement

The dairy plant under consideration produces a variety of end products which are specified by their product recipe and packaging type. We are given the following data regarding these product recipes and packaging types:

Z	the set of product recipes
W	the set of packaging types
$\lambda_z^{\text{proc}}/\lambda_w^{\text{pack}}$	the processing/packaging rates for every product recipe z/packaging type w
g_z	the standardization time for product recipe z
s_z	the dry-matter concentration of product recipe z

The production system involves facilities to process, store, and package evaporated milk products. The following data are available with respect to those facilities:

$\Pi^{\text{proc}}/\Pi^{\text{pack}}$	the set of processing/packaging lines
$\Pi_z^{\text{proc}}/\Pi_w^{\text{pack}}$	the set of processing/packaging lines which are suitable for processing/packaging product recipe z/packaging type w
$N^{\text{proc}}/N^{\text{pack}}$	the set of production cycles for processing/packaging lines
y	the number of storage tanks
q	the volume of a storage tank
$c^{\text{proc}}/c^{\text{pack}}/c^{\text{stor}}$	the cleaning time for processing/packaging/storage units
$h^{\text{proc}}/h^{\text{pack}}$	the maximum amount of time that processing/packaging lines can be used without cleaning

The production is driven by customer orders for end products. We are given the following data regarding the customer orders:

J	the set of customer orders
J_z/J_w	the set of customer orders concerning product recipe z/packaging type w
u_j	the size of customer order j

The problem is to find a production schedule minimizing the overall makespan. This requires to decide upon the specifications regarding material flows:

I	the set of standardization batches
I_z	the set of standardization batches concerning product recipe z
Γ_z	the matching of standardization batches to customer orders of recipe z, i.e. if $(i, j) \in \Gamma$, then standardization batch i serves customer order j

and the allocation of production resources and the timing of production operations:

$\Lambda^{\text{proc}}/\Lambda^{\text{pack}}$ the allocation of processing/packaging lines to standardization batches/customer orders, i.e. if $(i, p, n) \in \Lambda^{\text{proc}}$, then standardization batch i is processed in processing line p at cycle n, and if $(j, p, n) \in \Lambda^{\text{pack}}$ then customer order j is packaged in packaging line p at cycle n

$\upsilon_i^{\text{proc}}/\upsilon_i^{\text{stor}}/\upsilon_j^{\text{pack}}$ the time interval where standardization batch i/customer order j is processed/stored/packaged

while also respecting the constraints on sequence- and time-dependent cleaning requirements, and storage capacity limitations.

4.6 Mathematical model

In this section, we explain the modeling approaches taken in the matching and the scheduling phases. In these respective phases, we first determine the specifications of material flows, and then build the production schedule.

4.6.1 The matching phase

The objective in the matching phase is to find the minimum number of standardization batches that can serve all customer orders. Here, the main limitation is that the size of each standardization batch is bounded by the capacity of storage tanks. Thus, the total size of the customer orders served by the same standardization batch cannot exceed the capacity of a storage tank. We assume that the size of customer orders is smaller than the capacity of a storage tank. Nevertheless, if this is not the case, then large orders can be broken into several orders with smaller sizes.

It should be clear that all orders served by the same standardization batch must be of the same product recipe. Therefore, it is possible to determine the minimum number of standardization batches that can fill customer orders of a particular product recipe independent from other recipes. In the following, we formulate the matching problem of a particular product recipe as a MILP model.

Let us consider a particular recipe $z \in Z$, and the respective set of customer orders J_z. If each customer order were standardized as a batch, then the number of standardization batches would equal the number of customer orders. Thus, the cardinality of J_z is an upper bound on the number of standardization batches of product recipe z. Let $\tilde{I}_z$ be the set of prospective standardization batches, such that $|J_z| = |\tilde{I}_z|$. We define the following binary indicator variables:

$$\eta_i = \begin{cases} 1, & \text{if prospective standardization batch } i \text{ is active,} \\ 0, & \text{otherwise.} \end{cases}$$

$$\alpha_{ij} = \begin{cases} 1, & \text{if prospective standardization batch } i \text{ is used} \\ & \text{to serve customer order } j, \\ 0, & \text{otherwise.} \end{cases}$$

By means of these indicator variables, we can express the number of active standardization batches as $\sum_{i \in \tilde{I}_z} \eta_i$, and the size of prospective standardization batch i as $\sum_{j \in J_z} u_j \alpha_{ij}$. Then, we can represent the matching problem by the following MILP.

$$\min \sum_{i \in \tilde{I}_z} \eta_i \tag{4.1}$$

$$\sum_{j \in J_z} u_j \alpha_{ij} \leqslant q \eta_i \quad \forall i \in \tilde{I}_z \tag{4.2}$$

$$\sum_{i \in \tilde{I}_z} \alpha_{ij} = 1 \quad \forall j \in J_z \tag{4.3}$$

$$\eta_{i-1} \geqslant \eta_i \quad \forall i \in \tilde{I}_z : i > 1 \tag{4.4}$$

Eq. (4.1) sets the objective function which minimizes the number of active standardization batches. Eq. (4.2) guarantees that a standardization batch can only be used to serve customer orders if it is active, and if so, its size cannot exceed the capacity of a storage tank. Eq. (4.3) ensures that all customer orders are met. Eq. (4.4) enforces a numerical ordering of standardization batches for symmetry breaking purposes.

When solved for product recipe z, the aforementioned MILP yields the set of active standardization batches $I_z = \{i \mid i \in \tilde{I}_z \text{ and } \eta_i = 1\}$, and the set of matchings between batches and customer orders $\Gamma_z = \{(i, j) \mid i \in I_z, j \in J_z \text{ and } \alpha_{ij} = 1\}$. It is important to recall that the formulation is meant for a particular product recipe. When approaching the overall problem, we solve a matching problem for each product recipe to obtain all the necessary information regarding material flows.

4.6.2 The scheduling phase

We develop the scheduling model with the high-level modeling language ILOG OPL (Van Hentenryck, 1999) by using ILOG OPL Development Studio 6.0 (ILOG, 2008).

The efficiency of CP lies in its ability to effectively prune variable domains so that a large part of the search space does not have to be explored. The main tool employed in pruning the search space is the so-called constraint propagation. It relies on the idea of removing inconsistent values from variable domains which can be proven not to be a part of a feasible solution. This process is carried out by constraint-specific propagators which encapsulate efficient algorithms to deduce information in constraint propagation. Hence, an important factor concerning the computational performance of a CP model is the selection of constraint operators employed. Here, we make use of the special variables and operators ILOG OPL offers for scheduling purposes. These are explicitly mentioned in the text whenever necessary, and their semantics are provided in Appendix 4.A.

The building blocks of the proposed model are the interval variables expressing the timing of production tasks. An interval variable is defined by a set of attributes involving its start and end time, duration, and whether it is optional or not. These attributes may vary subject to the definition of the variable and the constraints of the model. The key modeling issues in the scheduling phase are the assignment of production tasks to production resources, and the configuration of production cycles. We model task assignments by using auxiliary tasks representing possible assignment options. Also, we use auxiliary tasks representing the start and end of production cycles. All the aforementioned tasks are modeled as interval variables as listed below:

$\upsilon_i^{\text{proc}}/\upsilon_i^{\text{stor}}/\upsilon_j^{\text{pack}}$	the interval variable representing the processing/storage/packaging task of standardization batch i/customer order j
$\upsilon_{ipn}^{\text{proc}}/\upsilon_{jpn}^{\text{proc}}$	the optional interval variable representing the auxiliary processing/packaging task of standardization batch i/customer order j being assigned to the processing/packaging line p at cycle n
$\tau_{pn}^{\text{proc}}/\tau_{pn}^{\text{pack}}$	the interval variable representing the start of cycle n on processing/packaging line p
$\varsigma_{pn}^{\text{proc}}/\varsigma_{pn}^{\text{pack}}$	the interval variable representing the end of cycle n on processing/packaging line p

It is important to remark that all υ_i^{proc} and υ_j^{pack} are defined with fixed durations since the amount of materials undergo those tasks and the respective production rates are known in advance. That is, for all standardization batches of recipe z, the length of υ_i^{proc} equals $\sum_{(i,j)\in\Gamma_z} u_j/\lambda_z^{\text{proc}}$, and for all customer orders of packaging type w, the length of υ_j^{pack} equals $u_j/\lambda_w^{\text{pack}}$. This is not the case for υ_i^{stor} since their lengths are determined in connection with the timing of processing and packaging tasks. The auxiliary variables $\upsilon_{ipn}^{\text{proc}}$ and $\upsilon_{jpn}^{\text{pack}}$ have equivalent lengths with the respective variables υ_i^{proc} and υ_j^{pack}. The variables $\tau_{pn}^{\text{proc}}/\tau_{pn}^{\text{pack}}$ and $\varsigma_{pn}^{\text{proc}}/\varsigma_{pn}^{\text{pack}}$ merely refer to points in time, and they are defined with null durations.

In what follows, we introduce the objective function and the constraints of the proposed CP model. For the sake of notational brevity, we first define the respective sets reflecting all possible processing and packaging assignments.

$$\tilde{\Lambda}^{\text{proc}} = \{(i,p,n) \mid z \in Z,\ i \in I_z,\ p \in \Pi_z^{\text{proc}} \text{ and } n \in N^{\text{proc}}\}$$
$$\tilde{\Lambda}^{\text{pack}} = \{(j,p,n) \mid w \in W,\ j \in J_w,\ p \in \Pi_w^{\text{pack}} \text{ and } n \in N^{\text{pack}}\}$$

Objective function. The objective function minimizes the overall makespan. We express the objective function in Eq. (4.5). The endOf operator simply returns the time instance where the underlying interval ends. Thus the objective function equals the latest end time among all packaging tasks.

$$\min \quad \max_{j \in J}\{\text{endOf}(\upsilon_j^{\text{pack}})\} \tag{4.5}$$

Timing constraints. Timing constraints guarantee that processing, packaging, and storage tasks are correctly synchronized. We make use of the operators startAtStart, endBeforeEnd, and endBeforeStart to specify these constraints (see Appendix 4.A). We express timing constraints in the following.

$$\text{startAtStart}(\upsilon_i^{\text{proc}}, \upsilon_i^{\text{stor}}) \quad \forall i \in I \tag{4.6}$$
$$\text{endBeforeEnd}(\upsilon_i^{\text{stor}}, \upsilon_j^{\text{pack}}, c^{\text{stor}}) \quad \forall z \in Z,\ \forall (i.j) \in \Gamma_z \tag{4.7}$$
$$\text{endBeforeStart}(\upsilon_j^{\text{pack}}, \upsilon_i^{\text{proc}}, g_z) \quad \forall z \in Z,\ \forall (i.j) \in \Gamma_z \tag{4.8}$$

Eq. (4.6) ensures that processing and storage tasks of a standardization batch start concurrently. Eq. (4.7) states that the packaging tasks of the customer orders served by the same standardization batch should end at least c^{stor} time units before the end of the storage task of the associated batch. Here, c^{stor} stands for the time required to clean a storage tank. Thus, rather than modeling the cleaning time of storage tanks explicitly, we extend the duration of storage tasks to cover the cleaning time. This approach significantly simplifies modeling the storage allocation. Eq. (4.8) guarantees that the necessary amount of time – which equals g_z for batches of product recipe z – is reserved for the standardization of batches before they are packaged.

Assignment constraints. Assignment constraints make sure that processing/packaging tasks are assigned to a particular cycle of a processing/packaging line. We use the $\mathrm{alternative}$ operator to express these constraints (see Appendix 4.A). Assignment constrains are given below.

$$\mathrm{alternative}(\upsilon_i^{\text{proc}}, \{\upsilon_{ipn}^{\text{proc}} \mid (i,p,n) \in \tilde{\Lambda}^{\text{proc}}\}) \quad \forall i \in I \tag{4.9}$$

$$\mathrm{alternative}(\upsilon_j^{\text{pack}}, \{\upsilon_{jpn}^{\text{pack}} \mid (j,p,n) \in \tilde{\Lambda}^{\text{proc}}\}) \quad \forall j \in J \tag{4.10}$$

Eq. (4.9) and Eq. (4.10) enforce that every processing and packaging task is assigned to a single cycle of a single production resource, and the optional interval variable corresponding to that assignment is active and synchronized with the interval variable representing the underlying task.

Sequencing constraints. Sequencing constraints state that production tasks that are carried out within the same cycle of a production line are sequenced in increasing order based on their dry-matter concentration. We use the $\mathrm{sequence}$ variable and the $\mathrm{noOverlap}$ operator to model sequencing constraints (see Appendix 4.A). We define $\Theta_{pn}^{\text{proc}}/\Theta_{pn}^{\text{pack}}$ as the sequence of optional interval variables that are carried out within the production cycle n of the processing/packaging line p. Thus, $\Theta_{pn}^{\text{proc}}/\Theta_{pn}^{\text{pack}}$ is a sequence of all $\upsilon_{ipn}^{\text{proc}}/\upsilon_{jpn}^{\text{proc}}$ with $p \in \Pi^{\text{proc}}/\Pi^{\text{pack}}$ and $n \in N^{\text{proc}}/N^{\text{pack}}$. We also introduce the $|Z| \times |Z|$ transition matrix V such that $V[z, z'] = M$ if $s_z < s_{z'}$ and 0 otherwise. Here, M refers to a large number. Then, we can write the sequencing constraints as follows.

$$\mathrm{noOverlap}(\Theta_{pn}^{\text{proc}}, V) \quad \forall n \in N^{\text{proc}},\ \forall p \in \Pi^{\text{proc}} \tag{4.11}$$

$$\mathrm{noOverlap}(\Theta_{pn}^{\text{pack}}, V) \quad \forall n \in N^{\text{pack}},\ \forall p \in \Pi^{\text{pack}} \tag{4.12}$$

Eq. (4.11) and Eq. (4.12) ensure the desired succession of tasks in processing and packaging units by imposing very large transition times when changing from a high concentrated recipe to a low concentrated one.

Storage constraints. Storage constraints guarantee that sufficient number or storage tanks are available to carry out storage operations. We model storage constraints by using a cumulative function expression (see Appendix 4.A). We define the cumulative function $F^{\text{stor}} = \sum_{i \in I} \mathrm{pulse}(\upsilon_i^{\text{stor}}, 1)$ to express the usage of storage tanks. Then, Eq. (4.13) guarantees that the number of storage tanks used simultaneously cannot exceed the number of available storage tanks – which is equal to y.

$$F^{\text{stor}} \leqslant y \tag{4.13}$$

Cycle constraints. Cycle constraints mediate the production cycles on processing and packaging lines. We make use of the precedence operators $\mathrm{endBeforeStart}$ and

$\mathrm{startBeforeEnd}$ to formulate these constraints (see Appendix 4.A). Below we articulate three types of cycle constraints.

$$\mathrm{endBeforeStart}(\tau_{pn}^{\mathrm{proc}}, \upsilon_{ipn}^{\mathrm{proc}}) \quad \forall (i,p,n) \in \tilde{\Lambda}^{\mathrm{proc}} \tag{4.14}$$

$$\mathrm{endBeforeStart}(\upsilon_{ipn}^{\mathrm{proc}}, \varsigma_{pn}^{\mathrm{proc}}) \quad \forall (i,p,n) \in \tilde{\Lambda}^{\mathrm{proc}} \tag{4.15}$$

$$\mathrm{endBeforeStart}(\tau_{pn}^{\mathrm{pack}}, \upsilon_{jpn}^{\mathrm{pack}}) \quad \forall (j,p,n) \in \tilde{\Lambda}^{\mathrm{pack}} \tag{4.16}$$

$$\mathrm{endBeforeStart}(\upsilon_{jpn}^{\mathrm{pack}}, \varsigma_{pn}^{\mathrm{pack}}) \quad \forall (i,p,n) \in \tilde{\Lambda}^{\mathrm{pack}} \tag{4.17}$$

Eq. (4.14) – Eq. (4.17) make sure that the optional interval variables reflecting the assignment of production tasks to particular cycles must lie within the time frame of the respective cycles.

$$\mathrm{startBeforeEnd}(\varsigma_{pn}^{\mathrm{proc}}, \tau_{pn}^{\mathrm{proc}}, -h^{\mathrm{proc}}) \quad \forall p \in \Pi^{\mathrm{proc}},\ \forall n \in N^{\mathrm{proc}} \tag{4.18}$$

$$\mathrm{startBeforeEnd}(\varsigma_{pn}^{\mathrm{pack}}, \tau_{pn}^{\mathrm{pack}}, -h^{\mathrm{pack}}) \quad \forall p \in \Pi^{\mathrm{pack}},\ \forall n \in N^{\mathrm{proc}} \tag{4.19}$$

Eq. (4.18) and Eq. (4.19) reflect the time-dependent cleaning requirements by restricting the length of processing and packaging cycles to the respective time limits h^{proc} and h^{pack}.

$$\mathrm{endBeforeStart}(\varsigma_{pn-1}^{\mathrm{proc}}, \tau_{pn}^{\mathrm{proc}}, c^{\mathrm{proc}}) \quad \forall p \in \Pi^{\mathrm{proc}},\ \forall n \in N^{\mathrm{proc}} : n > 1 \tag{4.20}$$

$$\mathrm{endBeforeStart}(\varsigma_{pn-1}^{\mathrm{pack}}, \tau_{pn}^{\mathrm{pack}}, c^{\mathrm{pack}}) \quad \forall p \in \Pi^{\mathrm{pack}},\ \forall n \in N^{\mathrm{pack}} : n > 1 \tag{4.21}$$

Eq. (4.20) and Eq. (4.21) provides a numerical ordering of cycles so as to guarantee that cycles follow each other based on a numerical succession.

4.7 Case study

In this section, we first provide the details of the evaporated milk production system which motivated this study, and then conduct a numerical study where we develop production schedules for several real-life cases by using the proposed approach. The data originates from a processing plant of a dairy company. However, they are slightly modified because of confidentiality concerns.

The plant manufactures over 200 evaporated milk products which differ in terms of product recipe and packaging type. Following customer orders, each week around 8 to 10 recipes are produced. Product recipes can be classified into three groups based on dry-matter concentration: low-, medium-, and high-concentrated recipes. The specifications of recipes, i.e. processing rate and standardization time, significantly differ from one recipe to another. The data regarding the most demanded 10 product recipes are provided in Table 4.1. There are two processing lines. The first one can handle all types of product recipes. The second one is only capable of processing low-concentrated recipes. Processing lines require cleaning after operating for 16 hours, and whenever they switch to a recipe with a lower concentration. The cleaning of a processing line takes 4 hours. The plant offers evaporated milk products with two different can sizes. Packaging rate is higher for the larger cans. The data regarding packaging rates are provided in Table 4.2. Packaging operations are carried out by four dedicated packaging lines – for each size two lines. Packaging lines require cleaning after operating for 72 hours, and whenever they switch to a recipe with a lower concentration. The cleaning of a packaging line takes 3 hours. There are eight storage tanks each with a capacity of 120 tons. These storage tanks can be used for any production recipe. However, they need to be cleaned each time they are used. It takes half an hour to clean a storage tank.

Recipe	Dry-matter concentration	Processing rate (tons/minute)	Standardization time (minutes)
R1	Low	0.45	150
R2	Low	0.30	250
R3	Low	0.30	250
R4	Low	0.25	150
R5	Low	0.40	250
R6	Medium	0.30	350
R7	Medium	0.30	400
R8	Medium	0.30	650
R9	Medium	0.40	500
R10	High	0.50	550

Table 4.1: Data for product recipes

Packaging type	Can size (grams)	Packaging rate (tons/minute)
C1	170	0.15
C2	410	0.25

Table 4.2: Data for packaging types

We consider 4 case examples. All these cases are known to be challenging because they required more than 6 days of production time following the manual scheduling approach in practice. They involve around 50 – 70 customer orders for products concerning 8 – 10 different product recipes and 2 different packaging types. The characteristics of the case examples are provided in Table 4.3. The historical data reveals that the plant employs around 6 – 7 cycles in the processing stage and 3 – 4 cycles in the packaging stage. Thus, in our application, we set the maximum number of cycles to reasonable limits of 10 and 6 for the processing and packaging stages respectively.

We implemented the model by using ILOG OPL Studio 6.0 modeling environment on an Intel i5 2.67 Ghz CPU platform with 4GB RAM. The MILP models in the matching phase are solved by using CPLEX 11.1, and the CP models in the scheduling phase is solved by employing CP 2.0. All computational runs are performed with a CPU time limit of 600 seconds. The computational results of the case examples are summarized in Table 4.3.

	Case 1	Case 2	Case 3	Case 4
Data				
# of customer orders	60	52	65	55
# of product recipes	10	8	8	9
# of packaging types	2	2	2	2
Total demand volume (tons)	3924	3688	4012	3902
Results of the matching phase				
# of standardization batches	40	32	41	37
Results of the scheduling phase				
# of processing cycles – max	7	6	7	6
# of packaging cycles – max	4	4	5	4
Makespan (minutes)	8069	7742	8191	7965

Table 4.3: Characteristics and computational results of case examples

It is important to note that the matching phase was solved to optimality in a marginal computational time in all case examples. However, optimality was not achieved in the scheduling phase of any case example within the given computational time limit. This is not unexpected given the complexity of the scheduling problem. Nevertheless, we observed that, for all case examples, the proposed method was able to find a feasible schedule with a makespan less than 6 days.

The results regarding the number of cycles are mostly in line with the data from the company. The best schedules found by the proposed model employs around 6 – 7 cycles for the processing stage and 3 – 4 cycles for the packaging stage. However, it appears that, as opposed to the traditional setting which often calls for 3 cycles, it may be a good option to employ at least 4 cycles in the packaging stage. This can mainly be attributed to the relatively short cleaning time required for the packaging stage.

For illustrative purposes we report the results of Case 1 in detail. The detailed order data for Case 1 are provided in Table 4.4. This case involves 60 customer orders with sizes varying between 10 and 120 tons. The total demand volume is around 4000 tons, and most of this demand (around 45% in volume) concerns a single product recipe labeled as R2.

The matching phase of the proposed approach yields 40 standardization batches for Case 1. These are listed together with their matchings with individual customer orders in Table 4.5. We observe that most of the standardization batches serve 2 customer orders. Nevertheless, there are also examples where a standardization batch serves 1 or 3 customer orders.

Figure 4.4 graphically illustrates the best schedule obtained for Case 1 by means of a Gantt chart with time on the horizontal and processing and packaging units on the vertical axis. The standardization batches and customer orders are represented with boxes shaded with three different intensities. These reflect the dry-matter concentration of the underlying product recipes.

A closer look at the Gantt chart reveals that the utilization is higher in the processing stage as compared to the packaging stage. We know that in general the processing rates are higher than the packaging rates. However, it appears that the advantage of having a larger number of units in the packaging stage overcomes the advantage of higher processing rates in the processing stage.

The intermediate storage tank usage profile corresponding to the best schedule is shown in Figure 4.5. As can be expected, the number of storage units used in the beginning and the end of the planning horizon is rather small, whereas it fluctuates around the limit the rest of the time. We also conducted a sensitivity analysis on the number of available storage tanks. We saw that decreasing the number of storage tanks even by a single unit significantly diminish performance. However, we observed that increasing the number of storage tanks has a marginal effect. It should be obvious that a simple sensitivity analysis is not sufficient to draw reliable conclusions. Nevertheless, it appears from the aforementioned observations that current storage capacity is probably sufficient to synchronize the two production stages. Notice that there are eight storage tanks available in the current setting. These can

#	Recipe	Package	Size (tons)	#	Recipe	Package	Size (tons)	#	Recipe	Package	Size (tons)
1	R1	C1	45	21	R2	C2	79	41	R5	C1	13
2	R1	C1	28	22	R2	C2	78	42	R5	C2	97
3	R1	C1	10	23	R2	C2	73	43	R5	C2	72
4	R1	C2	120	24	R2	C2	65	44	R5	C2	30
5	R1	C2	52	25	R2	C2	57	45	R6	C2	37
6	R1	C2	18	26	R2	C2	49	46	R7	C1	91
7	R2	C1	120	27	R2	C2	43	47	R7	C1	50
8	R2	C1	94	28	R2	C2	35	48	R7	C1	58
9	R2	C1	68	29	R2	C2	30	49	R7	C2	99
10	R2	C1	67	30	R2	C2	27	50	R8	C1	91
11	R2	C1	23	31	R2	C2	25	51	R8	C1	79
12	R2	C2	120	32	R2	C2	21	52	R8	C1	65
13	R2	C2	120	33	R3	C1	53	53	R8	C2	120
14	R2	C2	97	34	R4	C1	98	54	R8	C2	36
15	R2	C2	95	35	R4	C1	61	55	R9	C1	82
16	R2	C2	93	36	R4	C2	98	56	R9	C2	79
17	R2	C2	91	37	R4	C2	41	57	R9	C2	19
18	R2	C2	90	38	R5	C1	98	58	R9	C2	14
19	R2	C2	84	39	R5	C1	83	59	R10	C1	65
20	R2	C2	82	40	R5	C1	44	60	R10	C2	52

Table 4.4: Customer order data for Case 1

#	Recipe	Served orders	#	Recipe	Served orders
1	R1	4	21	R2	8, 32
2	R1	1, 5, 6	22	R3	33
3	R1	2, 3	23	R4	35, 37
4	R2	9, 28	24	R4	36
5	R2	16	25	R4	34
6	R2	11, 15	26	R5	41, 42
7	R2	12	27	R5	39, 44
8	R2	24, 29	28	R5	40, 43
9	R2	13	29	R5	38
10	R2	10, 27	30	R6	45
11	R2	18	31	R7	46
12	R2	22	32	R7	47, 48
13	R2	19	33	R7	49
14	R2	17, 31	34	R8	50
15	R2	25. 26	35	R8	51
16	R2	20	36	R8	53
17	R2	23, 30	37	R8	52, 54
18	R2	14	38	R9	55, 57, 58
19	R2	21	39	R9	56
20	R2	7	40	R10	59, 60

Table 4.5: Results of the matching phase for Case 1

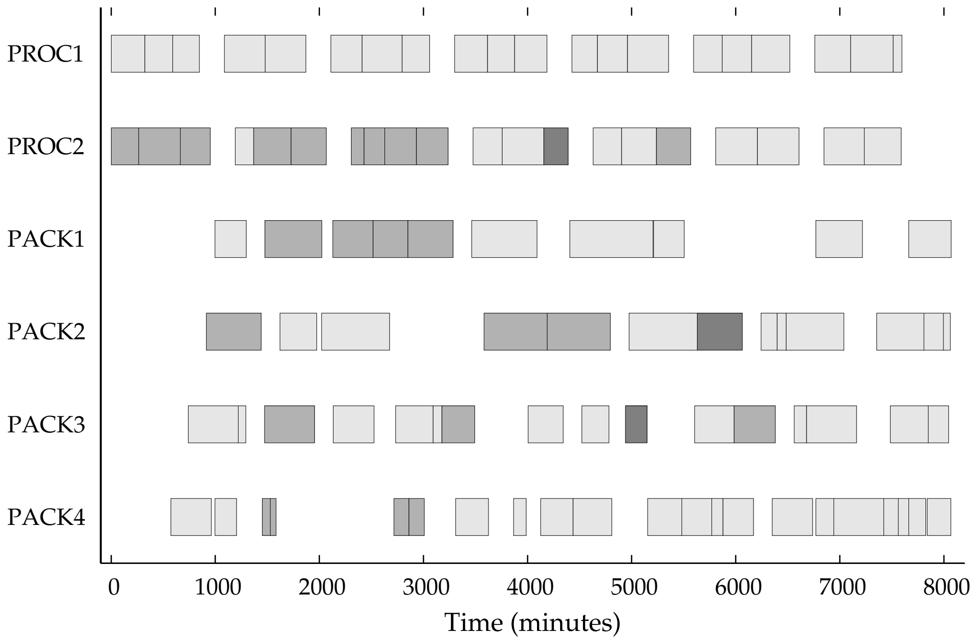

Figure 4.4: Gantt diagram illustrating the best solution for Case 1

simultaneously feed all four packaging lines, serve both two processing lines, and provide room for two batches for standardization.

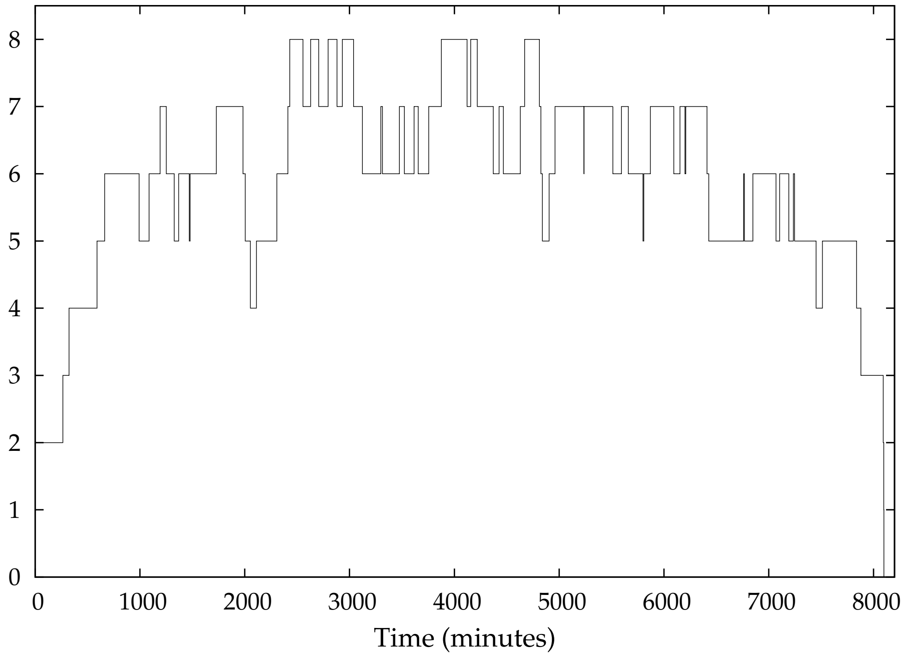

Figure 4.5: The storage tank usage profile for Case 1

The scheduling approach currently being used in the case company is mainly based on ad-hoc rules, and it is carried out by individual planners by hand. Thus, it is rather difficult to assess the performance of the proposed approach against manual scheduling in a one-to-one fashion. Nevertheless, the case examples show that the proposed approach looks very promising at the least.

4.8 Conclusions and extensions

In this study, we addressed a real-life scheduling problem encountered in a dairy plant specialized in evaporated milk products. The problem is computationally challenging, and it requires the consideration of the industry-specific characteristics of the underlying production environment. Therefore it necessitates an efficient and flexible modeling approach. We contribute to the literature by presenting such a mathematical approach which could be used in the context of a computer-aided scheduling system. The proposed approach comprises of two phases. In the first phase, it determines the specifications regarding material flows. Then, in the second phase, it builds a complete production schedule to realize the specified material flows. The isolation of material flow and scheduling decisions makes it straightforward to account for the traceability requirements which are critical in food processing industries. We applied the proposed approach in a case originated from a real-life evaporated milk plant, and observed that it significantly outperforms the manual scheduling approach currently in use.

The majority of scheduling research on food processing systems focuses on a particular production stage and ignores the effects of the respective scheduling decisions on the overall system performance. Thus, there is still a gap in the literature in coordinating local and global scheduling objectives. The current study can be regarded as an effort towards filling this gap by presenting an approach that reflects upon the system as a whole. We particularly targeted evaporated milk production processes. However, we addressed many characteristics which are common in food processing systems of make-and-pack configuration. Thus, the proposed approach can also be adapted to be employed in other production environments.

There are several interesting directions for further research. First, it is substantial to develop an optimal approach which can simultaneously handle the matching and the scheduling phases that are dealt with separately in the current study. It should be obvious that the practicality of such an approach is questionable since it would require a tremendous computational effort. However, it would provide a yardstick that could be used to assess the solution quality of alternative approaches. The adequacy of the proposed approach can be improved by addressing further elements which are of interest in the food processing industry. Rong and Grunow (2010) recently introduced the notion of chain dispersion in the context of traceability. They defined chain dispersion as a measure in which production batches are spread among different customers, and pointed out the importance of limiting the extent of dispersion to improve food safety. This concept can easily be embedded into the proposed approach by adopting the mathematical model employed in the matching phase. This could be done either by penalizing dispersion in the objective function, or by limiting dispersion by introducing a new set of constraints. The decomposition scheme employed in the current study simplified the overall problem significantly. This enabled us to find good solutions in reasonable computation times. Nevertheless, the mathematical problem considered in the scheduling phase is still very demanding. This is evident from the fact that we were not able solve the case problem to optimality. We employed a CP model in formulating the scheduling problem. This model also supports employing user-defined search procedures to specify how the search space is explored. Hence, more sophisticated search algorithms could be embedded into the proposed CP model in order to increase its effectiveness. Finally, an important research direction for further research is to extend the proposed approach to account for possible revisions in customer orders. This is a common issue in the food processing industry since customers tend to revise their orders before they are dispatched. There are two main approaches towards handling such fluctuations: safety stocks and safety times. As recently pointed out by Van Kampen et al (2010) safety time appears be more promising in make-to-order environments such as the one considered in the current study.

Appendix 4.A Semantics of CP operators

The CP formulation provided in Section 4.6.2 makes use of special scheduling operators of ILOG OPL modeling language. Here, we provide the semantics of these operators. We discuss the interval and sequence variables, cumulative function expressions, and related built-in constraint structures. The reader is referred to (ILOG, 2008) for the complete overview of the OPL modeling language.

Interval variables. Interval variables represent tasks or operations characterized by a start and an end time. An important feature of interval variables is that they can be optional. Thus based on the presence of an interval variable its execution can be modeled as a decision variable. The domain of an interval variable a, i.e. $\mathrm{dom}(a)$, is a subset of $\{\perp\} \cup \{[s,e) \mid s,e \in \mathbb{Z}_+, s \leqslant e\}$. For and interval variable a which is not optional $\perp \notin dom(a)$. The value of the interval variable a is denoted as $\underline{a}$. An interval variable a is absent if $\underline{a} = \perp$, and present if $\underline{a} = [s,e)$. We respectively denote the presence status, the start, and the end of $\underline{a}$ as $x(\underline{a})$, $s(\underline{a})$, and $e(\underline{a})$. If an interval variable a is absent then $x(\underline{a}) = 0$ and the start and end are undefined.

Sequence variables. Sequence variables represent the total ordering of a set of interval variables. Thus, an interval sequence variable p defined on a set interval variables A is a decision variable whose possible values are the permutations of the intervals of A. It is important to note that any absent interval variables are not considered in the sequence. Let $\underline{A}$ represent the values of a set of interval variables and n denote the cardinality of $\underline{A}$. A permutation π of $\underline{A}$ is a function $\pi : \underline{A} \rightarrow$

$\{\perp\} \cup [1, n]$, and has the properties that $\forall \underline{a} \in \underline{A} \Leftrightarrow (\pi(\underline{a}) \neq \perp)$ and $\forall \underline{a}, \underline{b} \in \underline{A}, (x(\underline{a}) \wedge x(\underline{b}) \wedge \underline{a} \neq \underline{b}) \Rightarrow (\pi(\underline{a}) \neq \pi(\underline{b}))$.

Cumulative function expressions. Cumulative function expressions represent the cumulated usage of renewable resources over time. A cumulative function expression f is an expression defined on $\mathbb{Z}$ to $\mathbb{Z}^+$. The individual contributions of interval variables to cumulative function expressions are described via elementary cumulative functions. For the purposes of this study we are only interested in the elementary function pulse. Let a be an interval variable and $h \in \mathbb{Z}$. Then, $\mathrm{pulse}(a, h)$ can be characterized by a function F such that $F(t) = h$ if $t \in [s(\underline{a}), e(\underline{a}))$, and $F(t) = 0$ otherwise. Here, h stands for the consumption of the cumulative resource by the activity represented by the interval variable a. Algebraic sums of elementary cumulative functions construct cumulative function expressions. They have the form $f = \sum_i \epsilon_i f_i$ where $\epsilon_i \in \{-1, +1\}$ and f_i is an elementary function expression.

Constraints on interval and sequence variables. Constraints on interval and sequence variables are used to confine the relative positioning of interval variables, create logical links between interval variables, and model disjunctive resources.

The $\mathrm{noOverlap}$ constraint on a sequence variable p states that the intervals comprised in p do not overlap. Furthermore, it specifies the minimum time gap that must separate consecutive intervals in p. Let A be the set of interval variables in the permutation π of p, and $\underline{A}$ represent their values. Also, let $T(a)$ denote the type of interval variable a, and V be the transition distance matrix defining the minimum distance between different types of interval variables. Then, the constraint $\mathrm{noOverlap}(p, V)$ is defined as:

$$\begin{aligned}\mathrm{noOverlap}(p, V) \Leftrightarrow \forall \underline{a}, \underline{b} \in \underline{A}, \neg x(\underline{a}) \vee \neg x(\underline{b}) \vee \\ ((\pi(\underline{a}) < \pi(\underline{b})) \Leftrightarrow (e(\underline{a}) + V[T(\pi, \underline{a}), T(\pi, \underline{b})] \leqslant s(\underline{b})))\end{aligned}$$

The $\mathrm{alternative}$ constraint models a specific statement with regard to the presence of a set of interval variables. Let $a, b_1, \ldots, b_n$ be interval variables. The constraint $\mathrm{alternative}(a, \{b_1, \ldots, b_n\})$ states that if interval a is present then exactly one of intervals $\{b_1, \ldots, b_n\}$ is present and synchronized with a. Thus, $\mathrm{alternative}$ constraint holds if and only if:

$$\neg x(\underline{a}) \Leftrightarrow \forall i \in [1,n]\ \neg x(\underline{b_i})$$
$$x(\underline{a}) \Leftrightarrow \exists k \in [1,n]\ \begin{cases} x(\underline{b_k}) \wedge (s(\underline{a}) = s(\underline{b_k})) \wedge (e(\underline{a}) = e(\underline{b_k})) \\ \forall j \in [1,n] \backslash \{k\}\ \neg x(\underline{b_j}) \end{cases}$$

The relative positioning of interval variables can be controlled by means of the precedence constraints $\mathrm{startAtStart}$, $\mathrm{endBeforeEnd}$, $\mathrm{endBeforeStart}$, and $\mathrm{startBeforeEnd}$. Let a and b be two interval variables, and z be an integer. Then, the aforementioned precedence constraints are defined as follows.

$$\begin{aligned} \mathrm{startAtStart}(a,b,z) &\Leftrightarrow x(\underline{a}) \wedge x(\underline{b}) \Rightarrow s(\underline{a}) + z = s(\underline{b}) \\ \mathrm{endBeforeEnd}(a,b,z) &\Leftrightarrow x(\underline{a}) \wedge x(\underline{b}) \Rightarrow e(\underline{a}) + z \leqslant e(\underline{b}) \\ \mathrm{endBeforeStart}(a,b,z) &\Leftrightarrow x(\underline{a}) \wedge x(\underline{b}) \Rightarrow e(\underline{a}) + z \leqslant s(\underline{b}) \\ \mathrm{startBeforeEnd}(a,b,z) &\Leftrightarrow x(\underline{a}) \wedge x(\underline{b}) \Rightarrow s(\underline{a}) + z \leqslant e(\underline{b}) \end{aligned}$$

Chapter 5

Order acceptance in food processing systems with random raw material requirements

This chapter is published as Kilic, Van Donk, Wijngaard, and Tarim (2010), Order acceptance in food processing systems with random raw material requirements, OR Spectrum, 32(4): 905 - 925.

Abstract. *This study considers a food production system that processes a single perishable raw material into several products having stochastic demands. In order to process an order, the amount of raw material delivery from storage needs to meet the raw material requirement of the order. However, the amount of raw material required to process an order is not exactly known beforehand as it becomes evident during processing. The problem is to determine the admission decisions for incoming orders so as to maximize the expected total revenue. It is demonstrated that the problem can be modeled as a single resource capacity control problem. The optimal policy is shown to be too complex for practical use. A heuristic approach is proposed which follows rather simple decision rules while providing good results. By means of a numerical study, the cases where it is critical to employ optimal policies are highlighted, the effectiveness of the heuristic approach is investigated, and the effects of the random resource requirements of orders are analyzed.*

5.1 Introduction

The food processing industry is characterized by divergent product structures where a small number of (agricultural) raw materials are used to produce a large variety of customer specific end products (e.g. Akkerman and Van Donk, 2009). Due to the large variety of end products it is often not possible or at least inefficient to produce all end products in make-to-stock fashion. Hence, make-to-order (MTO) is the typical production setting in the food processing industry.

Food processing industry involves highly perishable raw materials which are usually replenished periodically during their relatively short harvest seasons. The raw material procurement costs are relatively high compared to operational costs. Therefore, firms often face the issue of covering demands with limited amounts of raw materials being significantly of less value at the end of the season.

Another important characteristic of the food processing industry is the variability in production yield. This issue basically derives from two sources. First, the food processing industry involves raw materials whose qualities are often variable (Fransoo and Rutten, 1994). Most quality parameters, such as protein, fat, and sugar content are usually hard to measure reliably. Some others, such as texture, smell and taste can only be measured in a subjective way. Hence, it is hard to know the exact amount of raw material that is needed to process a given amount of end product (Somsen and Capelle, 2002). Second, the production process itself involves variability (Fransoo and Rutten, 1994; Flapper et al, 2002). The yield variability due to the production process is often associated with the type of the operation and production quantities involved (Murthy and Ma, 1996). The yield of a production run is affected by the inconsistencies in processing operations (involving chemical reactions), disturbances (i.e. starting up, changeovers, finishing), and packaging operations. Henceforth, either a part of the batch or all of it may fail to fulfill certain quality specifications and may need to be disposed of as waste or by-product. In such cases, additional production runs, and hence, additional raw materials are required. This issue is particularly important in MTO environments where demands are rigid and shortages are not allowed (Grosfeld-Nir and Gerchak, 2004).

In the production environments discussed above, an important planning problem is how to allocate available raw material to incoming orders over time according to their relative importance in order to achieve better operational performance (Fransoo and Rutten, 1994; Van Donk, 2000). This study is motivated by this practical and pervasive issue encountered in food processes.

A typical real-life example of the above mentioned problem can be found in the potato starch industry. The basic and most obvious process is the conversion of potatoes into starch during the harvest season. Starch is used in many different applications such as food, textile, paper, adhesives, and detergents. Given the size of the industry and technologies involved, products are made and marketed in different business units. Since the main aim of a potato starch company is to sell all starch during the year in order to get the highest value, it is common practice to allocate a certain amount of raw material to each business unit with the explicit demand to transform it into marketable products at the highest possible price. Some allocations depend upon the specific characteristics of the starch, for example, starch from modified potatoes. It should be obvious that there is a considerable drawback in still having inventory at the end of the year, if new potato starch is then available. Some products made out of potato starch are technologically advanced and highly customized, and they show a rather erratic almost lumpy demand pattern. In these cases product orders require specific types of processing operations usually involving chemical reactions, which lead to order-specific revenues and variable raw material requirements, often because the production process is insufficiently under control. Throughout the year, orders for various products arrive and they are subsequently either accepted or rejected, as a business unit aims at maximizing its returns. The admission decisions are given according to revenues and raw material requirements associated with incoming orders. The revenue gained by accepting a particular order is known at the time of making the acceptance decision. However, the raw material requirement of an order is not known with certainty due to the variability in yield. As already noted, given the nature of the company and its policies, and due to the relatively perishable nature of the raw material, the remaining inventory (if there is any) has to be disposed of at a low value, for example, as waste or a by-product. The key problem here is to establish decision rules that coordinate the admission decisions for incoming orders. Similar types of decision situations can be found in milk processing (where a certain amount has to be processed into products) and other food industries.

The problem we consider in this study falls into the category of capacity control problems in revenue management which have attracted great interest from both practitioners and researchers. Revenue management is used in situations where a finite amount of products/services have to be allocated to several classes of customer. The reader is referred to Talluri and Van Ryzin (2004) for an overview of this field. Revenue management literature offers a large variety of studies concentrating on establishing optimal policies for capacity control problems especially for airline management practices. Lee and Hersh (1993) is one of the first studies to characterize the structure of the optimal policy for the basic capacity control problem. They showed that the optimal policy is threshold-type relying on the remaining time and the remaining resource level. In other words, for a given resource level, a given order is accepted only if the remaining time is less than an order-specific time threshold; and similarly, for a given remaining time, a given order is accepted only if the remaining inventory is larger than an order-specific resource level threshold. However, these easily implementable threshold-type policies are only optimal when the resource requirements of orders are unit-sized as is the case for airline seat allocation problems (see e.g. Lee and Hersh, 1993; Papastavrou et al, 1996; Van Slyke and Young, 2000; Kleywegt and Papastavrou, 2001; Brumelle and Walczak, 2003). From a modeling point of view, order-specific resource requirements do not pose much difficulty. However, they have a profound impact on the structure of the optimal policy since the optimal expected revenue function no longer preserves some of the basic monotonicity properties (Talluri and Van Ryzin, 2004). In the case of non-unit resource requirements, the behavior of the optimal admission decision is rather complex, since the optimal policy is not threshold-type. As a result, practical use and implementation is limited, since a very careful and precise examination of the resource level throughout time is required.

The literature mentioned above provides a strong background for the problem we address. However, there are some specific characteristics of the food processing industry, such as, the random resource requirements of orders and disposal costs, which have not yet been addressed. In this study, we stylize and streamline the raw material allocation problem in the food processing industry by addressing the above mentioned characteristics. We build on the well-established revenue management models. We do not aim to characterize the optimal policy since it is known that it does not possess a simple structure. Rather, we are rather interested in: (i) pointing out the cases where it is critical and necessary to employ optimal/near-optimal admission policies; (ii) developing a heuristic approach possessing a rather simple structure while providing satisfactory performance; and (iii) analyzing the effects of the stochasticity of resource requirements of orders.

The remainder of this paper is organized as follows. In Section 5.2, we provide the formal problem definition. In Section 5.3, we present a dynamic program (DP) to compute the optimal policy. In Section 5.4, we discuss the structural properties of the optimal policy. In Section 5.5, we propose two simple and easily implementable heuristics for the problem. In Section 5.6, we conduct a numerical study and investigate the effects of different problem settings on the performance of the optimal policy and heuristics. Finally, in Section 5.7, we draw our conclusions and propose some extensions of the study.

5.2 Problem definition

Consider a food processing system where a key perishable resource (raw material) is used to process a set of order types indexed by $i = 1, \ldots, m$. The planning horizon is composed of t discrete time periods indexed by $n = 0, \ldots, t-1$. The resource inventory at the beginning of period 0 involves s units of material. The remaining resource at the end of period $t-1$ (or the fictitious period t) is disposed of as waste or by-product with a unit disposal cost of c. Customer orders arrive throughout the planning horizon. In each time period at most a single order may arrive. In time period n there is a probability p_{in} of a type-i order arrival and $p_{0n} = 1 - \sum_{i=1}^{m} p_{in} \geqslant 0$ of no order arrival. Orders are either accepted or rejected as a whole (i.e. complete admission). Upon the arrival of an order, its type and associated revenue become known. The revenue gained by accepting a type-i order is denoted as r_i. However, the resource requirement of a type-i order, denoted as w_i, is random as it emerges during processing the order. Hence, the decision maker accepts or rejects an incoming order without knowing the exact resource requirement. The resource requirement of a type-i order follows a known probability mass τ_i which depends on the process technology and product recipe used in processing type-i orders. When the resource inventory is insufficient to fulfill an accepted order, the shortage can be covered from an external source at a unit shortage (penalty) cost of z. We assume that z is large enough that it is not profitable to accept an order when there is no resource on hand. Otherwise, it would be optimal to accept all incoming orders. The resource inventory is reviewed throughout the planning horizon and the decision maker knows the resource level when an order arrives.

The admission decisions depend on: (i) arrival processes, profitabilities, and resource requirements of incoming orders; (ii) the current resource level and the time remaining until the end of the planning horizon; and (iii) associated cost parameters. The basic intuition is that, when the resource level is low and the time remaining until the end of the planning horizon is long, it would be reasonable to reject less profitable orders in order to preserve resources so as to be able to accept more profitable future orders. Also, for high levels of raw material and a short planning horizon, accepting every order seems reasonable. We analyze the admission policies, characterized by decision rules for given resource levels, and time periods for accepting/rejecting orders, maximizing expected revenue accumulated throughout the planning horizon.

5.3 Dynamic program

The problem we define in Section 5.2 can be modeled as a dynamic program. Let $g_n(\cdot)$ be the optimal revenue function at period n, that is, $g_n(x)$ represents the expected revenue if the initial resource level is x units and the optimal admission decisions are made throughout the rest of the planning horizon. Then, we can write:

$$g_n(x) = \sum_{i=1}^{m} p_{in} v_{in}(x) + p_{0n} g_{n+1}(x) \tag{5.1}$$

where

$$v_{in}(x) = \max\{r_i + \mathrm{E}\left[g_{n+1}(x - w_i)\right], g_{n+1}(x)\} \tag{5.2}$$

with the terminal revenues

$$g_t(x) = -c(x)^+ - z(x)^- \tag{5.3}$$

incorporating both disposal and penalty costs where $(x)^+ := \max\{0, x\}$ and $(x)^- := \max\{0, -x\}$. As stated in the problem definition, the penalty cost of shortage is independent of time and z is large enough to prevent any acceptance decision when there is no resource available. These enable us to express the penalty cost in the terminal revenue function and set $g_n(x) = -zx$ for all $x \leqslant 0$.

It is clear that an optimal policy for the dynamic program accepts a type-i order when the resource level is x in period n only if

$$r_i \geqslant g_{n+1}(x) - \mathrm{E}\left[g_{n+1}(x - w_i)\right]. \tag{5.4}$$

The left-hand side of Eq. (5.4) represents the immediate incremental revenue, and the right-hand side is the expected loss in future revenue by accepting a type-i order.

In order to solve the DP, one needs to compute $g_n(x)$ for $x = 1, \ldots, s$ and $n = 0, \ldots, t-1$ by backward recursion. Note that there is no need to evaluate $g_n(x)$ for $x \leqslant 0$ explicitly since they are all equal to $-zx$.

5.4 Structural properties

In this section, we explore some monotonicity properties of the DP given in Section 5.3 which may result in acceptance policies that are easy to implement. We begin our discussion by considering a simple version of the problem. Let us assume that each order is characterized by the same constant unit-sized resource requirement, that is, $w_i = 1$ for all $i = 1, \dots, m$. Furthermore, let us assume that the disposal cost is zero and there is no shortage option. In this case, the problem is reduced to the basic seat allocation problem in airline revenue management with multiple demand classes. It can be shown that the optimal revenue function possesses some important structural properties (see e.g. Lee and Hersh, 1993). For this problem, Eq. (5.4) can be re-written as

$$r_i \geqslant g_{n+1}(x) - g_{n+1}(x-1). \tag{5.5}$$

Notice that now the expected loss in future revenue by accepting an order is independent of the order type. Based on this observation, Lee and Hersh (1993) state the main properties of $g_n(x)$ by means of the following theorem:

Theorem 5.4.1 (Lee and Hersh (1993)). *When all orders have unit-sized resource requirements,*

1. $g_n(x) - g_n(x-1)$ *is non-increasing in* x *for any given* n*,*
2. $g_n(x) - g_n(x-1)$ *is non-increasing in* n *for any given* x*.*

Theorem 5.4.1 shows that the expected loss in future revenue by accepting an order (or the marginal value of an additional unit of resource) is higher when the resource level is relatively low and/or the remaining time until the end of the planning horizon is relatively long. The monotonicity of $g_n(x)$ leads to the following implications:

1. For each order type i and any given period n there exists a critical resource level x^* satisfying $r_i \geqslant g_{n+1}(x) - g_{n+1}(x-1)$ for all $x \geqslant x^*$ such that a type-i order is rejected whenever $x < x^*$ and accepted otherwise.
2. For each order type i and any given resource level x there exists a critical time period n^* satisfying $r_i \geqslant g_{n+1}(x) - g_{n+1}(x-1)$ for all $n \geqslant n^*$ such that a type-i order is rejected whenever $n < n^*$ and accepted otherwise.

Let us illustrate these results by means of a simple numerical example.

Example 5.4.1. *Consider a five-period problem with two order types (Type-1 and Type-2) both having unit-sized resource requirements. The arrival probabilities of order types are stationary over time and they are both equal to 0.5, that is,* $p_{1n} = p_{2n} = 0.5$ *for all*

$n = 0, \ldots, 4$. The respective rewards of orders are $r_1 = 1$ and $r_2 = 2$. Since there are only two order types to be considered, each with unit-sized resource requirements, it is clear that the more profitable order type, that is, Type-2, would be accepted whenever there are sufficient resources available (i.e. $x \geqslant 1$). However, Type-1 orders can be rejected in order to allocate the available resources for Type-2 order arrivals in later periods.

We evaluate initial resource levels $[0, 5]$. Since there are only five periods in each of which at most one order can arrive, the initial resource level of 5 units is the maximum amount of resources that could possibly be used. Table 5.1 presents the optimal expected rewards and optimal admission decisions corresponding to each decision period and resource level.

x/n	0		1		2		3		4	
5	7.500	$(1,1)$	6.000	$(1,1)$	4.500	$(1,1)$	3.000	$(1,1)$	1.500	$(1,1)$
4	6.469	$(0,1)$	6.000	$(1,1)$	4.500	$(1,1)$	3.000	$(1,1)$	1.500	$(1,1)$
3	5.281	$(0,1)$	4.938	$(0,1)$	4.500	$(1,1)$	3.000	$(1,1)$	1.500	$(1,1)$
2	3.781	$(0,1)$	3.625	$(0,1)$	3.375	$(0,1)$	3.000	$(1,1)$	1.500	$(1,1)$
1	1.969	$(0,1)$	1.938	$(0,1)$	1.875	$(0,1)$	1.750	$(0,1)$	1.500	$(1,1)$
0	0.000	$(0,0)$	0.000	$(0,0)$	0.000	$(0,0)$	0.000	$(0,0)$	0.000	$(0,0)$

Rows and columns stand for the respective resource level x and the decision period n and each entry presents the expected revenue and the admission decisions (1 = acceptance and 0 = rejection) for Type-1 and Type-2 orders, respectively.

Table 5.1: The optimal expected revenues and admission decisions for Example 5.4.1

Let us consider Type-1 orders which are less preferable as compared to Type-2 orders. The critical resource levels of Type-1 orders are $5, 4, 3, 2, 1$ in periods $0, 1, 2, 3, 4$ respectively. For the given periods, Type-1 orders are accepted only when the resource level is higher than the critical level. The critical time periods for Type-1 orders are $4, 3, 2, 1, 0$ for resource levels $1, 2, 3, 4, 5$ respectively. For the given resource levels, Type-1 orders are accepted only when the time period is later than the critical time period. The relationship between the acceptance decisions and the optimal average rewards can also be observed in Table 5.1. For instance, consider the last period, that is Period 4. Here, any incoming order would be accepted as long as $x \geqslant 1$. Hence, the optimal expected reward equals $0.5 \times 1 + 0.5 \times 2 = 1.5$ for all $x \geqslant 1$ and 0 for $x = 0$. Then, we can say that the marginal value of an additional resource at $n = 4$ is 0 for all $x \geqslant 1$ and 1.5 for $x = 0$. Consequently, since the revenue gained by accepting a Type-1 order is equal to 1, a Type-1 order at period 3 is accepted for all $x \geqslant 2$ and rejected for all $x < 2$.

The above example illustrates the threshold-type behavior of the optimal policy.

However, the problem addressed in this paper possesses a number of additional features which may affect this behaviour. Below, we investigate whether the simple threshold policies apply to the problem we address.

First, we consider the non-zero disposal costs and shortage penalty costs. Introducing non-zero disposal costs and shortage penalty costs leads to a terminal reward $g_t(x) = -c(x)^+ - z(x)^-$ which is linearly decreasing on $x \geqslant 0$ with rate c and linearly increasing on $x \leqslant 0$ with rate z. Henceforth, in this case, $g_t(x)$ is a concave function of x which preserves the property that $g_t(x) - g_t(x-1)$ is non-increasing in x. This shows that non-zero disposal costs and shortage penalty costs do not interfere with the critical resource levels and decision periods. Hence, the optimal policies are still threshold-type.

Second, we consider non-unit resource requirements. When requirements are not unit sized, the behavior of the optimal policy is rather complex. In general, the monotonicity properties of $g_n(x)$ discussed so far do not hold in case of non-unit resource requirements (see e.g. Lee and Hersh, 1993; Brumelle and Walczak, 2003). This is due to the combinatorial behavior of the problem which derives from the large variety of options to match available resources with the resource requirements of orders of different types. We illustrate the effect of non-unit resource requirements with a simple example.

Example 5.4.2. *Consider again the problem explained in Example 5.4.1. Here we make a simple change in the parameter values of order Type-2. Let the resource requirement of Type-2 orders w_2 be 2 units rather than 1, and let the revenue gained by fulfilling a Type-2 order be 4 rather than 2. We leave the rest of the parameters unchanged.*

We evaluate initial resource levels $[0, 10]$. Since $w_2 = 2$, the initial resource level of 10 units is the maximum amount of resource that could possibly be used in this example. Table 5.2 presents the optimal expected rewards and optimal admission decisions corresponding to each decision period and resource level.

It is easy to observe that the critical resource levels are non-existent in this case. Let us consider the admission decisions regarding Type-1 orders in period 0. The optimal admission decision here is to accept Type-1 orders at resource levels $1, 3, 5, 7, 8, 9, 10$, and to reject them at resource levels $2, 4, 6$. Hence, there is no critical resource level for Type-1 orders. It is easy to interpret this result. When the resource level is an odd number, after allocating the available resource to more profitable orders, that is, Type-2 orders which have a resource requirement of 2 units, the remaining 1 unit of slack resource can only be allocated to Type-1 orders. This is a simple illustration of

x/n	0		1		2		3		4	
10	12.500	$(1,1)$	10.000	$(1,1)$	7.500	$(1,1)$	5.000	$(1,1)$	2.500	$(1,1)$
9	12.375	$(1,1)$	10.000	$(1,1)$	7.500	$(1,1)$	5.000	$(1,1)$	2.500	$(1,1)$
8	11.906	$(1,1)$	10.000	$(1,1)$	7.500	$(1,1)$	5.000	$(1,1)$	2.500	$(1,1)$
7	10.938	$(1,1)$	9.750	$(1,1)$	7.500	$(1,1)$	5.000	$(1,1)$	2.500	$(1,1)$
6	9.969	$(0,1)$	9.063	$(1,1)$	7.500	$(1,1)$	5.000	$(1,1)$	2.500	$(1,1)$
5	8.313	$(1,1)$	7.813	$(1,1)$	7.000	$(1,1)$	5.000	$(1,1)$	2.500	$(1,1)$
4	7.344	$(0,1)$	6.875	$(0,1)$	6.125	$(1,1)$	5.000	$(1,1)$	2.500	$(1,1)$
3	4.875	$(1,1)$	4.750	$(1,1)$	4.500	$(1,1)$	4.000	$(1,1)$	2.500	$(1,1)$
2	3.906	$(0,1)$	3.813	$(0,1)$	3.625	$(0,1)$	3.250	$(0,1)$	2.500	$(1,1)$
1	0.969	$(1,0)$	0.938	$(1,0)$	0.875	$(1,0)$	0.750	$(1,0)$	0.500	$(1,0)$
0	0.000	$(0,0)$	0.000	$(0,0)$	0.000	$(0,0)$	0.000	$(0,0)$	0.000	$(0,0)$

Rows and columns stand for the respective resource level x and the decision period n and each entry presents the expected revenue and the admission decisions (1 = acceptance and 0 = rejection) for Type-1 and Type-2 orders, respectively.

Table 5.2: The optimal expected revenues and admission decisions for Example 5.4.2

matching available resources to the resource requirements of different order types. It is not profitable to preserve e.g. the first, the third, or the fifth unit of resource for possible future orders of Type-2. Hence they should be allocated to orders of Type-1.

It is also possible to show that the optimal policy possesses an irregular behavior with respect to the remaining decision periods. The interested reader is referred to Brumelle and Walczak (2003) for further examples illustrating this type of irregularity.

Example 5.4.2 clearly shows that the optimal admission policy presents an irregular behavior with respect to the available resource level in case of non-unit resource requirements. In other words, the optimal admission decision regarding an order type may switch from acceptance to rejection and then from rejection to acceptance a number of times on the resource level axis given a decision period. Implementing such a policy in practice is very difficult, since it would require a careful examination of the resource level upon arrival of an order over time. There has been some work on characterizing the special cases where optimal admission policies are still threshold-type (see e.g. Papastavrou et al, 1996; Brumelle and Walczak, 2003). For example, if splitting orders is allowed (i.e. partial admission), then the optimal policy is still threshold-type. However, these special cases are rather restrictive, and do not hold for the problem addressed in this study.

Finally, we will discuss the stochasticity of resource requirements. So far we have not explicitly considered this specific characteristic of the problem we address in this study. Nonetheless, the discussion provided in this section can be generalized to the problem with stochastic requirements. The problem with deterministic resource requirements is a special case of the problem with random resource requirements. Consequently, we know that the optimal revenue function of the stochastic problem shows an irregular behavior as in the deterministic case. Thus the non-optimality of the simple threshold-type policies also applies to the problem with stochastic resource requirements.

Taken all together, these observations show that it is fairly easy to model and solve the resource allocation problem in food processes via standard approaches from the literature. However, the resulting policies are rather complex and difficult to implement in practice.

5.5 Heuristic approaches

We have shown that the optimal admission policy of the problem under consideration does not have an easily implementable structure. In this section, we propose two heuristic approaches which follow simple decision rules and, therefore, can easily be implemented in practice. In the following subsections we provide the details of these approaches which we refer to as two-band heuristic and first-come-first-served heuristic.

5.5.1 The two-band heuristic

The two-band (TB) heuristic limits the irregular behavior of the optimal policy and provides simple decision rules regarding resource levels. The underlying intuition of the TB heuristic is based on two simple arguments:

1. It must be profitable to accept an order when the resource level is "sufficiently high" that it is not necessary to preserve resources for future orders with higher rewards.
2. It must be profitable to accept an order when the resource level is "sufficiently low" that it is not possible to accept future orders with higher rewards because of their larger resource requirements.

Henceforth, one can think of two bands on the resource level axis for each order type such that an incoming order is accepted whenever the resource level lies within one of these bands. We refer to those bands as the higher and the lower acceptance bands. Each band can be characterized by the critical resource levels setting its upper and lower bounds. In other words, the higher and lower acceptance bands of order type-i in period n involves the respective resource levels within $[\underline{x}_{in}^{\text{high}}, \overline{x}_{in}^{\text{high}}]$ and $[\underline{x}_{in}^{\text{low}}, \overline{x}_{in}^{\text{low}}]$. Since there are only two acceptance bands along the resource levels axis, the resulting admission policy under the TB heuristic is very simple. The admission decision regarding a given order type only switches at the boundaries of the two acceptance bands and remains the same for all other resource levels.

When those bounds characterizing the higher and the lower acceptance bands are known, the revenue function of the TB heuristic $g_n^{\text{TB}}(x)$ can be written as:

$$g_n^{\text{TB}}(x) = \sum_{i=1}^{m} p_{in} v_{in}^{\text{TB}}(x) + p_{0n} g_{n+1}^{\text{TB}}(x) \tag{5.6}$$

where

$$v_{in}^{\text{TB}}(x) = \begin{cases} r_i + \mathrm{E}\left[g_{n+1}^{\text{TB}}(x - w_i)\right] & \text{if } \underline{x}_{in}^{\text{low}} \leqslant x \leqslant \overline{x}_{in}^{\text{low}} \text{ or } \underline{x}_{in}^{\text{high}} \leqslant x \leqslant \overline{x}_{in}^{\text{high}} \\ g_{n+1}^{\text{TB}}(x) & \text{otherwise} \end{cases} \tag{5.7}$$

with the terminal revenue function given in Eq. (5.3).

We design the TB heuristic as a simplified version of the optimal policy, which ignores most of the irregularities in the optimal revenue function. Hence, the critical resource levels can be obtained by a simple search procedure within the backward recursion used for the optimal DP. Since the higher acceptance band corresponds to sufficiently high resource levels, we can assume that $\overline{x}_{in}^{\text{high}} = \infty$. In other words, type-$i$ orders are accepted in period n whenever the resource level is higher than $\underline{x}_{in}^{\text{high}}$. The remaining bounds $\underline{x}_{in}^{\text{high}}$, $\underline{x}_{in}^{\text{low}}$, and $\overline{x}_{in}^{\text{low}}$ are then obtained by means of the following equations.

$$\underline{x}_{in}^{\text{high}} = \sup\left\{x + 1 : r_i < \mathrm{E}\left[g_{n+1}^{\text{TB}}(x - w_i)\right] - g_{n+1}^{\text{TB}}(x)\right\} \tag{5.8}$$

$$\underline{x}_{in}^{\text{low}} = \inf\left\{x : r_i \geqslant \mathrm{E}\left[g_{n+1}^{\text{TB}}(x - w_i)\right] - g_{n+1}^{\text{TB}}(x)\right\} \tag{5.9}$$

$$\overline{x}_{in}^{\text{low}} = \inf\left\{x - 1 : r_i < \mathrm{E}\left[g_{n+1}^{\text{TB}}(x - w_i)\right] - g_{n+1}^{\text{TB}}(x), x > \underline{x}_{in}^{\text{low}}\right\} \tag{5.10}$$

There are three possible cases regarding the existence of the acceptance bands for a given order type i and decision period n: (i) both acceptance bands exist, that is, there are at least two non-consecutive resource levels where an order will be accepted ($\underline{x}_{in}^{\text{high}} > \overline{x}_{in}^{\text{low}}$); (ii) only a single acceptance band exists, that is, there is at least one block of resource levels where an order will be accepted ($\underline{x}_{in}^{\text{high}} = \underline{x}_{in}^{\text{low}}$); and (iii) neither the upper nor the lower band exist, that is, there is no resource level where an order will be accepted ($\{x : r_i < \mathrm{E}[g_{n+1}^{\text{TB}}(x - w_i) - g_{n+1}^{\text{TB}}(x)\} = \emptyset$).

For all periods, the search procedure systematically evaluates resource levels in terms of the condition $r_i \geqslant \mathrm{E}[g_{n+1}^{\mathrm{TB}}(x - w_i)] - g_{n+1}^{\mathrm{TB}}(x)$ which specifies whether accepting a type-i order leads to a non-negative increment in the expected revenue. The procedure first checks the existence of acceptance bands. Then, it sets the upper and lower bounds of the higher and lower bands from the initial resource level downwards and from 0 upwards, respectively. At each iteration the procedure also computes the revenue function of the TB heuristic following Eq. (5.6) and Eq. (5.7).

We illustrate the basic principles of the TB heuristic by means of a simple example.

Example 5.5.1. *Let us consider again the problem sketched in Example 5.4.2 and use the TB heuristic rather than the optimal policy. Table 5.3 presents the expected rewards and admission decisions corresponding to each decision period and each resource level.*

x/n	0		1		2		3		4	
10	12.500	(1,1)	10.000	(1,1)	7.500	(1,1)	5.000	(1,1)	2.500	(1,1)
9	12.375	(1,1)	10.000	(1,1)	7.500	(1,1)	5.000	(1,1)	2.500	(1,1)
8	11.906	(1,1)	10.000	(1,1)	7.500	(1,1)	5.000	(1,1)	2.500	(1,1)
7	10.938	(1,1)	9.750	(1,1)	7.500	(1,1)	5.000	(1,1)	2.500	(1,1)
6	9.969	(0,1)	9.063	(1,1)	7.500	(1,1)	5.000	(1,1)	2.500	(1,1)
5	8.250	(0,1)	7.813	(1,1)	7.000	(1,1)	5.000	(1,1)	2.500	(1,1)
4	7.344	(0,1)	6.875	(0,1)	6.125	(1,1)	5.000	(1,1)	2.500	(1,1)
3	4.813	(0,1)	4.688	(0,1)	4.500	(1,1)	4.000	(1,1)	2.500	(1,1)
2	3.906	(0,1)	3.813	(0,1)	3.625	(0,1)	3.250	(0,1)	2.500	(1,1)
1	0.969	(1,0)	0.938	(1,0)	0.875	(1,0)	0.750	(1,0)	0.500	(1,0)
0	0.000	(0,0)	0.000	(0,0)	0.000	(0,0)	0.000	(0,0)	0.000	(0,0)

Rows and columns stand for the respective resource level x and the decision period n and each entry presents the expected revenue and the admission decisions (1 = acceptance and 0 = rejection) for Type-1 and Type-2 orders, respectively.

Table 5.3: The expected revenues and admission decisions for Example 5.5.1

We can observe the acceptance bands in all decision periods characterizing the TB heuristic. For example, in period 0, for Type-1 orders, the higher acceptance band involves resource levels $[7, \infty)$ and the lower acceptance band involves the resource level of 1 unit. When compared to the results regarding the optimal policy (see Table 5.2) we can see that the admission policy of the TB heuristic is more stable. However, there is also a loss in the expected revenue due to not making the optimal admission decisions. Consider period 0 with the initial resource level of 5 units. If the optimal admission policy is implemented then the expected revenue will be 8.313, whereas, if the heuristic admission policy is implemented then the expected revenue will be 8.250.

5.5.2 The first-come-first-served heuristic

The first-come-first-served (FCFS) heuristic is an approach where incoming orders are attended to in the sequence they arrive. It thus addresses the case where no action is taken to ration the resource inventory. Henceforth, it is a logical benchmark to gauge the effectiveness of any allocation policy. In our numerical analysis, we use the FCFS heuristic to assess the optimal policy and the TB heuristic.

The optimal admission policy depends on the revenues gained as immediate results of the acceptance/rejection actions and the expected revenues associated with ensuing resource levels. The FCFS heuristic considers only the former and accepts any incoming order which will lead to an immediate nonnegative increment in the expected revenue. That is, a type-i order is accepted in period n if $r_i \geqslant z\,\mathrm{E}\left[(x-w_i)^-\right]$. Note that the decision rule is independent of the possible revenues associated with the subsequent periods. Furthermore, it does not rely on the period in which the decision is made. Hence, it can be translated into a static acceptance threshold for each order type. That is, the FCFS policy accepts a type-i order only if the resource level is higher than an order specific threshold. We denote this threshold by $\underline{x}_i$. We can then write:

$$\underline{x}_i = \inf\left\{x : r_i \geqslant z\,\mathrm{E}\left[(x-w_i)^-\right]\right\}. \tag{5.11}$$

Consequently, the revenue function of the FCFS heuristic $g_n^{\text{FCFS}}(x)$ can be written as:

$$g_n^{\text{FCFS}}(x) = \sum_{i=1}^{m} p_{in} v_{in}^{\text{FCFS}}(x) + p_{0n} g_{n+1}^{\text{FCFS}}(x), \tag{5.12}$$

where

$$v_{in}^{\text{FCFS}}(x) = \begin{cases} r_i + \mathrm{E}\left[g_{n+1}^{\text{FCFS}}(x-w_i)\right] & \text{if } x \geqslant \underline{x}_i \\ g_{n+1}^{\text{FCFS}}(x) & \text{otherwise} \end{cases} \tag{5.13}$$

with the terminal revenue function given in Eq. (5.3).

5.6 Numerical study

We conduct numerical studies in order to analyze:

1. The performances of the FCFS and the TB heuristics
2. The effects of considering the stochasticity of resource requirements
3. The effects of penalty and disposal costs

5.6.1 Experiment settings

As we discussed, the complex structure of the optimal admission policy is due to the combinatorial behavior of the problem. This behavior derives from the large variety of options to match available resources with the resource requirements of orders of different types. Hence, in our numerical study our aim is to investigate cases with a variety of order types and parameters characterizing different system configurations.

An arbitrary problem instance is characterized by a set of parameters: rewards r_i, probability mass functions τ_i, and arrival probabilities p_{in} associated with each order type i; the number of periods t, the initial inventory level s, the penalty cost z, and the disposal cost c. We assume that the resource requirements of orders follow a discretized truncated normal distribution. We characterize the stochasticity of the resource requirements by a coefficient of variation ρ common for all order types. Notice that this assumption enables us to uniquely characterize the probability mass of each order type τ_i for a given mean $\overline{w}_i$ and a coefficient of variation ρ.

We consider three main classes of random instances involving $m \in \{2, 5, 10\}$ types of orders. For each of these classes, we generate four sub-classes by imposing the coefficient of variation levels $\rho \in \{0.00, 0.05, 0.15, 0.25\}$. For each sub-class, we randomly generate 10^3 instances with various rewards r_i, average resource requirements $\overline{w}_i$, and arrival probabilities p_{in}. The rewards and average resource requirements of each order type are selected from the set $\{10, 20, \ldots, 100\}^2$ with uniform probability. The arrival probabilities are assumed to be stationary over time, that is, $p_{in} = p_i$. The probability of no order arrival p_0 is set to 0.2. The arrival probability of each order is selected from $(0, 1)$ with uniform probability. They are then normalized such that the arrival probabilities of orders and the no arrival probability sum up to 1. The number of periods t is fixed at 20.

For all instances we set the number of periods and compute the expected revenues of both the optimal policy and the heuristic approaches g, g^{FCFS}, and g^{TB} (we omit indices for simplicity's sake). In order to characterize the respective performances of the FCFS and the TB heuristics we define $\Delta^{\text{TB}} = (1 - g^{\text{TB}}/g) \times 100$ and $\Delta^{\text{FCFS}} = (1 - g^{\text{FCFS}}/g) \times 100$.

We consider the initial inventory levels $x \in [0, 2\xi]$ where ξ is the expected total resource requirement, i.e. $\xi = \sum_{i=1}^{m} \sum_{n=0}^{t-1} p_{in}\overline{w}_i$. It is hard, however, to reflect all x values within the given range individually. Hence, rather than reporting the expected revenue for each x, we report the average expected revenues for a set of x values. In order to do so, we divide the whole domain $[0, 2\xi]$ into 20 intervals with equal lengths each covering 10% of the domain. For each interval $k \in \{1,\dots,20\}$ we report the average g, g^{FCFS}, and g^{TB} for $(k-1)\xi/20 < x \leqslant k\xi/20$. The number of random instances sums up to 4×10^3 for each sub-class and to 12×10^3 in total. We believe that this broad collection of instances should allow us to address some practical cases found in real-life applications.

Having generated a collection of random instances, we can now investigate the performance of any admission policy given a penalty cost z and a disposal cost c.

5.6.2 Numerical results and insights

In what follows, we discuss our findings in detail regarding the points raised at the beginning of this section.

The performances of the FCFS and the TB heuristics

In this sub-section we analyze the performances of the FCFS and the TB heuristics with respect to the optimal policy. We conduct a set of experiments considering all random instances with the respective penalty and disposal costs $z = 10$ and $c = 0.5$.

For all resource level intervals $k \in \{1,\dots,20\}$ we report on g, Δ^{TB}, and Δ^{FCFS}. The results can be found in Table 5.4, 5.5 and 5.6. These results show that g, Δ^{FCFS}, and Δ^{TB} are severely affected by the resource level, the coefficient of variation, and the number of order types.

To start with, it is interesting to examine the behavior of the revenue function g. From our discussion we know that g is not necessarily concave on the resource level. Nevertheless, we can observe that g first tends to increase and then to decrease with increasing resource levels in general. It is obvious that the optimal policy is more selective in accepting orders for resource levels where g tends to increase. Thus employing admission policies is mainly critical when resource levels are low.

The behavior of g is also reflected in the performance of the heuristic approaches. Both Δ^{FCFS} and Δ^{TB} tend to decrease with increasing resource levels. That is, the importance of making the optimal admission decisions decreases with increasing resource levels. One exception is the case with extremely low resource levels. Then Δ^{FCFS} and Δ^{TB} may increase moving from the resource level interval $k = 1$ to $k = 2$ (see e.g. Table 5.4). This result is also intuitive since the number of order types for which sufficient resources can be provided is very limited for those resource levels. As a result, the optimal policy in this case cannot be very selective in accepting orders. It is important to note that the TB heuristic is very competitive for all resource levels with a maximum Δ^{TB} of 0.96%.

Obviously the expected revenue decreases with the degree of stochasticity of the resource requirements of orders, regardless of the policy employed. Since the optimal policy is the one best suited to handle stochasticity, one may expect that it will perform relatively better than the other heuristics for high ρ values. However, the numerical results show that both Δ^{FCFS} and Δ^{TB} decrease with increasing ρ, especially for low resource levels. This result shows that the optimal policy is not robust with respect to the degree of stochasticity whereas both the FCFS and the TB heuristics are. This is a rather interesting result in the sense that simple control rules perform relatively better when a complicating factor such as stochasticity is higher.

It is clear that the optimal policy is more selective when the number of order types is large, since this leads to a variety of options to preserve resources for more profitable orders. This can be observed by considering the performance of the FCFS policy which does not preserve resources for future orders. Regardless of the resource level or the degree of the stochasticity, Δ^{FCFS} increases with the number of order types. In contrast to the FCFS heuristic, the performance of the TB heuristic improves as the number of order types increases. This can be clarified by considering the structure of the TB heuristic. The gap between the optimal policy and the TB heuristic stems from the irregular behavior of the revenue function which is mostly neglected by the TB heuristic. This irregular behavior arises because of the dissimilarity of the order types in terms of revenues and resource requirements. Note that the order types are picked randomly from a bounded set. As a consequence the similarity between them increases with the number of order types. Thus larger number of order types result in a more lenient optimal policy and thus positively affects the performance of the TB heuristic.

The effects of considering the stochasticity of resource requirements

We analyze here what happens if we ignore the stochasticity of resource requirements and follow the admission decisions tailored to the set of instances with deterministic resource requirements (i.e. $\rho = 0.00$) for instances characterized by stochastic resource requirements (i.e. $\rho \in \{0.05, 0.15, 0.25\}$). Here, we only consider the class of random instances involving five order types for simplicity's sake, since the results are analogous with the other sub-classes. We use the respective penalty and disposal costs $z = 10$ and $c = 0.5$. We only consider the initial inventory levels corresponding to $k = \{1, \ldots, 5\}$, since the importance of stochasticity becomes

	$\rho = 0$			$\rho = 0.05$			$\rho = 0.15$			$\rho = 0.25$		
k	g	Δ^{FCFS}	Δ^{TB}	g	Δ^{FCFS}	Δ^{TB}	g	Δ^{FCFS}	Δ^{TB}	g	Δ^{FCFS}	Δ^{TB}
20	473.57	0.00	0.00	473.57	0.00	0.00	473.57	0.00	0.00	473.55	0.00	0.00
19	517.27	0.00	0.00	517.27	0.00	0.00	517.27	0.00	0.00	517.19	0.00	0.00
18	560.96	0.00	0.00	560.96	0.00	0.00	560.95	0.00	0.00	560.81	0.00	0.00
17	604.62	0.00	0.00	604.62	0.00	0.00	604.59	0.00	0.00	604.38	0.00	0.00
16	648.20	0.00	0.00	648.19	0.00	0.00	648.13	0.00	0.00	647.81	0.00	0.00
15	691.55	0.01	0.00	691.54	0.01	0.00	691.40	0.01	0.00	690.85	0.01	0.00
14	734.25	0.02	0.00	734.21	0.02	0.00	733.85	0.02	0.00	732.75	0.02	0.00
13	774.67	0.06	0.00	774.53	0.06	0.00	773.42	0.06	0.00	770.83	0.07	0.00
12	806.21	0.18	0.01	805.84	0.18	0.01	803.18	0.19	0.00	797.94	0.21	0.00
11	816.16	0.58	0.05	815.30	0.58	0.02	810.84	0.58	0.01	803.36	0.60	0.00
10	795.15	1.62	0.12	794.37	1.58	0.06	789.52	1.52	0.01	781.54	1.51	0.01
9	751.99	3.50	0.23	750.57	3.44	0.12	745.18	3.32	0.02	737.68	3.22	0.01
8	694.37	6.33	0.31	693.23	6.24	0.16	688.21	6.01	0.03	681.13	5.78	0.01
7	630.52	9.73	0.37	629.21	9.65	0.19	624.20	9.36	0.04	617.39	9.03	0.01
6	560.80	13.52	0.42	559.44	13.45	0.22	554.21	13.13	0.04	547.52	12.69	0.02
5	482.39	17.60	0.48	481.39	17.50	0.26	476.79	17.04	0.05	470.28	16.50	0.02
4	396.37	21.29	0.56	395.20	21.25	0.32	390.08	20.85	0.06	383.73	20.17	0.02
3	296.91	24.95	0.71	296.05	24.85	0.41	291.77	24.08	0.08	285.69	23.12	0.03
2	183.56	26.13	0.96	182.63	26.09	0.59	178.23	25.25	0.15	172.65	23.72	0.06
1	49.62	14.57	0.45	49.14	14.51	0.36	46.38	13.26	0.24	42.44	11.33	0.23

Table 5.4: The optimal expected reward and relative errors of heuristic approaches averaged over the class of random instances involving 2 order types with $z = 10$ and $c = 0.5$

	$\rho = 0$			$\rho = 0.05$			$\rho = 0.15$			$\rho = 0.25$		
k	g	Δ^{FCFS}	Δ^{TB}	g	Δ^{FCFS}	Δ^{TB}	g	Δ^{FCFS}	Δ^{TB}	g	Δ^{FCFS}	Δ^{TB}
20	475.86	0.00	0.00	475.86	0.00	0.00	475.86	0.00	0.00	475.85	0.00	0.00
19	519.85	0.00	0.00	519.85	0.00	0.00	519.85	0.00	0.00	519.78	0.00	0.00
18	563.83	0.00	0.00	563.83	0.00	0.00	563.82	0.00	0.00	563.69	0.00	0.00
17	607.78	0.00	0.00	607.77	0.00	0.00	607.75	0.00	0.00	607.53	0.00	0.00
16	651.62	0.00	0.00	651.61	0.00	0.00	651.54	0.00	0.00	651.17	0.00	0.00
15	695.13	0.01	0.00	695.11	0.01	0.00	694.90	0.01	0.00	694.18	0.01	0.00
14	737.51	0.03	0.00	737.44	0.03	0.00	736.86	0.03	0.00	735.35	0.04	0.00
13	776.28	0.09	0.00	776.08	0.09	0.00	774.66	0.09	0.00	771.59	0.11	0.00
12	805.36	0.29	0.01	804.95	0.29	0.01	802.19	0.30	0.00	796.97	0.34	0.00
11	816.16	0.89	0.02	815.51	0.89	0.01	811.51	0.90	0.01	804.56	0.93	0.00
10	804.62	2.35	0.05	803.80	2.34	0.03	799.16	2.30	0.01	791.58	2.27	0.01
9	775.14	5.04	0.08	774.20	5.02	0.05	769.36	4.89	0.01	761.80	4.72	0.01
8	733.92	8.91	0.13	732.89	8.88	0.07	727.95	8.63	0.02	720.54	8.28	0.01
7	683.45	13.47	0.18	682.35	13.44	0.10	677.32	13.07	0.03	670.05	12.54	0.01
6	623.85	18.35	0.26	622.66	18.30	0.15	617.56	17.83	0.04	610.47	17.13	0.02
5	554.33	23.35	0.37	553.08	23.30	0.21	547.96	22.72	0.05	541.06	21.84	0.02
4	473.25	28.36	0.51	471.97	28.30	0.29	466.90	27.61	0.07	460.30	26.54	0.03
3	377.07	33.18	0.67	375.84	33.12	0.41	370.97	32.27	0.10	364.74	30.91	0.04
2	258.51	36.28	0.88	257.44	36.25	0.57	253.05	35.07	0.16	247.50	33.05	0.07
1	93.34	26.20	0.60	92.83	26.25	0.43	90.27	24.90	0.21	86.76	22.22	0.15

Table 5.5: The optimal expected reward and relative errors of heuristic approaches averaged over the class of random instances involving 5 order types with $z = 10$ and $c = 0.5$

	$\rho = 0$			$\rho = 0.05$			$\rho = 0.15$			$\rho = 0.25$		
k	g	Δ^{FCFS}	Δ^{TB}	g	Δ^{FCFS}	Δ^{TB}	g	Δ^{FCFS}	Δ^{TB}	g	Δ^{FCFS}	Δ^{TB}
20	459.91	0.00	0.00	459.91	0.00	0.00	459.91	0.00	0.00	459.91	0.00	0.00
19	503.70	0.00	0.00	503.70	0.00	0.00	503.70	0.00	0.00	503.64	0.00	0.00
18	547.47	0.00	0.00	547.47	0.00	0.00	547.46	0.00	0.00	547.35	0.00	0.00
17	591.22	0.00	0.00	591.21	0.00	0.00	591.19	0.00	0.00	590.99	0.00	0.00
16	634.87	0.00	0.00	634.86	0.00	0.00	634.78	0.00	0.00	634.41	0.00	0.00
15	678.15	0.01	0.00	678.12	0.01	0.00	677.88	0.01	0.00	677.10	0.01	0.00
14	720.07	0.02	0.00	719.98	0.02	0.00	719.31	0.03	0.00	717.65	0.04	0.00
13	757.84	0.09	0.00	757.62	0.09	0.00	756.08	0.11	0.00	752.86	0.13	0.00
12	785.78	0.33	0.00	785.37	0.33	0.00	782.62	0.36	0.00	777.46	0.40	0.00
11	797.07	1.02	0.01	796.48	1.02	0.01	792.66	1.04	0.00	785.96	1.08	0.00
10	789.08	2.62	0.02	788.37	2.63	0.01	783.97	2.60	0.01	776.60	2.57	0.00
9	765.44	5.55	0.02	764.66	5.55	0.02	760.04	5.44	0.01	752.56	5.26	0.01
8	730.17	9.65	0.03	729.34	9.65	0.02	724.64	9.41	0.01	717.21	9.03	0.01
7	684.61	14.37	0.05	683.74	14.36	0.03	679.02	14.01	0.01	671.72	13.43	0.01
6	629.02	19.32	0.07	628.13	19.32	0.04	623.43	18.86	0.02	616.32	18.09	0.01
5	563.08	24.36	0.10	562.17	24.36	0.06	557.54	23.81	0.02	550.69	22.85	0.01
4	485.68	29.45	0.15	484.75	29.45	0.09	480.26	28.79	0.03	473.75	27.62	0.02
3	393.79	34.38	0.23	392.89	34.40	0.14	388.65	33.60	0.05	382.64	32.14	0.03
2	279.78	38.14	0.34	278.98	38.19	0.23	275.26	37.11	0.09	270.05	35.03	0.05
1	114.17	28.99	0.40	113.75	29.12	0.32	111.57	27.83	0.19	108.42	24.98	0.15

Table 5.6: The optimal expected reward and relative errors of heuristic approaches averaged over the class of random instances involving 10 order types with $z = 10$ and $c = 0.5$

negligible for higher resource levels. The results related to this set of experiments are given in Table 5.7.

One obvious observation is that as ρ increases the gap between the stochastic and deterministic approaches gradually increases. In addition to this, for all policies, the gap between the deterministic and stochastic approach is relatively higher when the resource level is lower. This is due to the fact that the variation with respect to the total resource requirements of all prospective orders is lower than the sum of the variations of each prospective order. This is usually referred to as the pooling effect.

We can also observe that ignoring the uncertainty results in relatively larger losses for the heuristic approaches as compared to the optimal policy. Thus, especially when a heuristic approach is being used one should be certain that the stochasticity in material requirements is correctly accounted for.

The effects of penalty and disposal costs

Finally, we analyze the effects of penalty and disposal costs. Here, we only consider the sub-class of random instances involving five order types for simplicity's sake. Nevertheless, we would like to note that the results are very similar for the other sub-classes. We first fix the number of periods at 20 and consider the initial inventory levels corresponding to $k = \{1, \ldots, 5\}$. In order to analyze the effect of penalty cost, we fix the disposal cost at 0.5, and consider the penalty costs $\{10, 12, 14\}$. Similarly, in order to analyze the effect of disposal costs, we fix the penalty cost at 10, and consider the disposal costs $\{0.5, 1.0, 2.0\}$. We compute and report on the expected revenues of all proposed policies, that is, g, g^{FCFS}, and g^{TB}. The results can be seen in Tables 5.8 and 5.9.

The effects of penalty and disposal costs on the proposed policies are rather straightforward. As can be observed, increasing penalty and disposal costs negatively effects the expected rewards of all proposed policies. This effect is stronger when the resource level is rather low. Furthermore, the effects of those cost parameters are more severe when the stochasticity of resource requirements is larger.

5.7 Conclusions and extensions

We addressed the problem of determining the order acceptance/rejection decisions in a food processing system where a single raw material is processed into a set of different orders. We considered some specific characteristics of the food process-

		$\rho = 0.05$			$\rho = 0.15$			$\rho = 0.25$		
		g	g^{FCFS}	g^{TB}	g	g^{FCFS}	g^{TB}	g	g^{FCFS}	g^{TB}
5	STOCH	553.08	424.22	551.93	547.96	423.46	547.70	541.06	422.87	540.95
	DET	552.86	423.96	549.57	546.16	418.03	541.90	536.61	407.65	529.44
4	STOCH	471.97	338.39	470.59	466.90	337.99	466.59	460.30	338.13	460.17
	DET	471.73	338.11	467.54	464.92	332.11	458.92	455.52	321.73	445.53
3	STOCH	375.84	251.34	374.29	370.97	251.25	370.60	364.74	251.98	364.60
	DET	375.61	251.06	370.15	368.91	245.11	360.18	359.84	234.72	345.45
2	STOCH	257.44	164.11	255.97	253.05	164.30	252.66	247.50	165.69	247.33
	DET	257.27	163.81	250.43	251.23	157.78	239.29	242.98	147.31	223.23
1	STOCH	92.83	68.47	92.43	90.27	67.79	90.08	86.76	67.48	86.63
	DET	92.79	68.36	89.84	89.63	65.10	84.55	84.86	59.82	76.67

Table 5.7: The expected rewards of the optimal policy and heuristic approaches: The comparison of policies computed for the stochastic and deterministic resource requirements cases both applied in the stochastic resource requirements case. Results are averaged over the class of random instances involving 5 order types with $z = 10$ and $c = 0.5$

k	z	$\rho = 0$: g	$\rho = 0$: g^{FCFS}	$\rho = 0$: g^{TB}	$\rho = 0.05$: g	$\rho = 0.05$: g^{FCFS}	$\rho = 0.05$: g^{TB}	$\rho = 0.15$: g	$\rho = 0.15$: g^{FCFS}	$\rho = 0.15$: g^{TB}	$\rho = 0.25$: g	$\rho = 0.25$: g^{FCFS}	$\rho = 0.25$: g^{TB}
	10	554.33	424.87	552.40	553.08	424.22	552.02	547.96	423.46	547.77	541.06	422.87	541.01
5	12	553.60	424.00	551.44	552.19	423.55	551.08	546.58	423.11	546.38	541.06	422.45	539.13
	14	553.08	423.59	551.03	551.52	423.17	550.38	545.49	422.83	545.28	537.68	421.90	537.62
	10	473.25	339.04	470.94	471.97	338.39	470.67	466.90	337.99	466.67	460.30	338.13	460.24
5	12	472.45	338.15	469.82	471.00	337.73	469.64	465.44	337.75	465.19	460.30	337.89	458.26
	14	471.88	337.76	469.42	470.28	337.39	468.87	464.29	337.58	464.03	456.74	337.47	456.67
	10	377.07	251.97	374.62	375.84	251.34	374.38	370.97	251.25	370.68	364.74	251.98	364.67
5	12	376.16	251.06	373.26	374.76	250.68	373.21	369.38	251.11	369.08	364.74	251.93	362.56
	14	375.50	250.68	372.90	373.95	250.37	372.37	368.14	251.01	367.83	360.96	251.62	360.87
	10	258.51	164.73	256.30	257.44	164.11	256.06	253.05	164.30	252.75	247.50	165.69	247.42
5	12	257.37	163.81	254.60	256.14	163.46	254.68	251.25	164.25	250.93	247.50	165.80	245.07
	14	256.54	163.44	254.23	255.18	163.18	253.69	249.85	164.23	249.52	243.30	165.61	243.21
	10	93.34	68.89	92.89	92.83	68.47	92.55	90.27	67.79	90.20	86.76	67.48	86.74
5	12	91.65	67.76	90.64	91.05	67.40	90.75	88.11	66.89	88.04	86.76	66.37	84.16
	14	90.43	67.09	90.02	89.75	66.78	89.45	86.48	66.23	86.41	82.19	65.28	82.17

Table 5.8: The expected rewards of the optimal policy and heuristic approaches: The comparison of policies with respect to different levels of penalty costs $z \in \{10, 12, 14\}$. Results are averaged over the class of random instances involving 5 order types with $c = 0.5$

k	c	$\rho = 0$			$\rho = 0.05$			$\rho = 0.15$			$\rho = 0.25$		
		g	g^{FCFS}	g^{TB}	g	g^{FCFS}	g^{TB}	g	g^{FCFS}	g^{TB}	g	g^{FCFS}	g^{TB}
	0.5	554.33	424.87	552.40	553.08	424.22	552.02	547.96	423.46	547.77	541.06	422.87	541.01
5	1.0	544.16	419.42	541.01	542.61	418.76	540.90	536.45	417.56	536.17	528.32	416.10	528.26
	2.0	527.87	408.52	521.29	525.77	407.83	522.35	517.90	405.75	517.39	507.96	402.54	507.87
	0.5	473.25	339.04	470.94	471.97	338.39	470.67	466.90	337.99	466.67	460.30	338.13	460.24
4	1.0	464.52	334.04	460.84	462.94	333.37	460.88	456.88	332.56	456.53	449.17	331.94	449.09
	2.0	450.43	324.03	442.86	448.28	323.33	444.23	440.51	321.72	439.90	431.14	319.55	431.03
	0.5	377.07	251.97	374.62	375.84	251.34	374.38	370.97	251.25	370.68	364.74	251.98	364.67
3	1.0	369.77	247.22	365.87	368.26	246.59	366.00	362.47	246.16	362.07	355.30	246.21	355.21
	2.0	357.88	237.73	349.78	355.83	237.08	351.46	348.44	235.99	347.73	339.78	234.66	339.66
	0.5	258.51	164.73	256.30	257.44	164.11	256.06	253.05	164.30	252.75	247.50	165.69	247.42
2	1.0	252.72	160.21	249.33	251.43	159.59	249.36	246.27	159.45	245.85	239.93	160.22	239.83
	2.0	243.18	151.18	235.94	241.45	150.54	237.51	234.87	149.76	234.17	227.29	149.28	227.15
	0.5	93.34	68.89	92.89	92.83	68.47	92.55	90.27	67.79	90.20	86.76	67.48	86.74
1	1.0	89.23	64.60	88.47	88.65	64.18	88.23	85.74	63.25	85.65	81.80	62.44	81.77
	2.0	82.06	56.04	79.61	81.34	55.60	80.09	77.82	54.16	77.66	73.17	52.34	73.13

Table 5.9: The expected rewards of the optimal policy and heuristic approaches: The comparison of policies with respect to different levels of disposal costs $c \in \{0.5, 1.0, 2.0\}$. Results are averaged over the class of random instances involving 5 order types with $z = 10$

ing industry, such as random raw material requirements of orders, shortage penalty costs and disposal costs which have not yet been addressed in the literature. Our contribution is three-fold. First, we showed that the problem can be modeled and solved as a dynamic program. Second, since the optimal admission policy does not follow simple decision rules, we provided a heuristic approach, which we referred to as the TB heuristic, based on intuitive decision rules which can obtain good results. Third, with an extensive numerical study we examined the effects of various parameters on admission policies and pointed out those cases where it is critical to employ admission policies.

The main conclusions of our numerical study can be summarized as follows. We compared the optimal policy with the FCFS heuristic in order to see how critical it is to employ the optimal policy. We observed that employing the optimal admission policy is essentially important in case of limited resource levels. Obviously, when the initial inventory level can be set freely, there is hardly any need to use an admission policy since it will be optimal to accept most of the orders. We also saw that the penalty of not using the optimal policy is higher when there is a larger number of order types with a lower degree of stochasticity in their resource requirements. We observed that the overall performance of the TB approach is very good. The relative gap between the optimal policy and the TB heuristic quickly narrows down as the resource level increases. Also, the TB heuristic performs relatively better in cases characterized by a large number of order types and a high degree of stochasticity of the resource requirements of orders. We saw that considering the stochasticity of the resource requirements is very critical especially when heuristic approaches are being used. We also observed that the effects of disposal and penalty costs are larger when the degree of stochasticity of the resource requirements of orders is higher.

There are two directions for further research worth exploring. First, the production capacities and lead-times could be considered. Our model neglects the production side of the system. As a result, our results do not readily apply to cases where production capacities are limited and/or lead-times are not negligible. It would be specifically interesting to consider a case where during processing an order (with unknown material consumption), other orders might arrive that have to be accepted or rejected without exactly knowing the resource level. Second, the model can be extended for systems involving multiple raw materials. Referring to the analogy with revenue management problems, this case corresponds to capacity control problems with multi-leg flights.

Chapter 6

Conclusions and directions for further research

Abstract. *The final chapter presents the research findings of the thesis, discusses the conducted research and points out some directions for future research.*

This thesis is concentrated on planning and scheduling in the process industry. The work carried out in the thesis stands apart from the existing literature by considering industry-specific characteristics of processing systems. A specific emphasis is laid on the combinations several industry-specific characteristics and the interactions between them. The main research objectives of the thesis are defined as: (i) to contribute to the development of mathematical models that can be used as decision aids in scheduling processing systems with industry-specific characteristics, and (ii) to provide some insight into the order acceptance function in the process industry with respect to limitations in raw material availability. The thesis is organized as a collection of research papers which attend to these research objectives. The first research objective is confronted in Chapter 2, Chapter 3 and Chapter 4. These chapters addressed particular scheduling problems originating from specific production environments and developed models and methods thereof. The second objective is confronted in Chapter 5. This chapter considered the order acceptance problem in a processing system subject to limited raw material availability and variable raw material quality.

The research papers included in this thesis addressed and elaborated mathematical problems inspired by specific production environments. These problems differ from each other in terms of their technological specifications and managerial objectives. The extent to which the proposed models and methods are applicable in other production environments strongly depends on the degree of contingency between the characteristics of the proposed approaches and the underlying production environments. This thesis exercised both case-specific and general approaches. For example, Chapter 3 adopted the general discrete time formulation of batch processes which is not tailor-made for a specific application. Chapter 4, however, presented an approach building on the structure of the underlying production environment. It is obvious that case-specific approaches are more efficient than general approaches since they make use of the specific characteristics of the targeted production environments. However, often they are not easily applicable in problems originating from different production environments. In the following, the results are summarized and directions for further research are suggested for each individual chapter included in the thesis.

Chapter 2 addressed a scheduling problem confronted in two-stage food processing systems with production and storage capacity limitations. The problem also entails the use of flexible product recipes, and involves the selection of a set of intermediates and end product recipes characterizing how those selected intermediates are blended into end products. In this regard, the problem is an extension of the production lot scheduling problem with integrated design decisions. A comprehensive mixed integer linear programming (MILP) model is developed for the problem which integrates the decisions regarding production schedule as well as the specifications of intermediates and end products with the objective to minimize total operational costs. The model is then applied to a data set collected from a real-life case. The results derived from the numerical study are assessed to better understand the dynamics of the problem. It is shown that cost parameters and capacity limitations have significant effects on the production schedule and the selection of product recipes. It is observed that the fixed production setup costs and inventory holding costs can be regulated by altering the number of intermediates and end product recipes as well as the production schedule. Also, the trade-offs between capacity limitations and operational costs are investigated. It is shown that the limitations on production and storage capacities interact with each other. Thus, whether a particular type of capacity limitation is binding depends on its relative magnitude. It is observed, for the case example, that the storage capacity limitation is binding whereas the processing and the blending capacities are not. On this account, the extent of possible cost reductions that can be achieved by expanding the storage capacity is investigated. The results showed that the cost reduction due to an extra storage unit is not decreasing on the actual storage capacity because of the interactions between the decisions on the selection of intermediates and the scheduling of production operations. This points out that a careful investigation is required when expanding the storage capacity. Although the research carried out in this chapter particularly targets the food processing industry, it takes into account several characteristics common in many other processing systems. Hence, it could be possible to adopt the proposed model to be used in different production environments. A possible example could be the batch processing counterpart of the flow processing system addressed in this chapter. Also, the proposed approach can be extended by relaxing some of its restrictive assumptions. The problem is analyzed under a common cycle scheduling policy. This policy is widely used in practice due to its simplicity. Nevertheless, it is known that it may perform badly in some production settings. Thus, the problem can be analyzed under more sophisticated scheduling policies. Another restriction is that the storage units are assumed to be identical and they can only be assigned to a single type of intermediate. However, in many practical cases storage units are different in terms of their volume, and it may be possible to use storage units for several types of materials. Therefore, it is important to direct further research efforts towards these cases.

Chapter 3 is concentrated on the detailed short-term scheduling problem in multiproduct/multipurpose batch processes. The work carried out in this chapter extends the conventional discrete time MILP formulation for scheduling batch processes by introducing storage capacity and storage time limitations. These limitations are very common in many industries involving perishable intermediates and end products. A mathematical model is developed for the problem which is shown to be capable of handling various storage configurations involving single/multiple and dedicated/multipurpose storage vessels. By means of a numerical study it is illustrated that storage capacity and storage time limitations have significant effects on production and storage operations and significantly degrade the cost performance of batch processing systems. Also, it is shown that these effects can be averted to some extent by means of using multipurpose storage vessels. The proposed approach builds on the general discrete time formulation of batch processes. There is a variety of studies based on this formulation which aim at capturing different characteristics of batch process scheduling problems. Therefore, the proposed model can easily be extended by employing methods already suggested in the literature. Among those extensions the integration of sequence- and frequency-dependent setups, the use of time-based objective functions, and the application of reformulations designed to enhance the computational performance can be mentioned. An important direction for further research is to adopt the proposed batch process model to account for continuous and semi-continuous processes where storage limitations addressed in this chapter are of concern. Also, research efforts can be directed towards modeling the problem by using a continuous time formulation. Although continuous time formulations have various drawbacks, in principle they are more realistic since they yield more precise solutions.

Chapter 4 investigated a real-life scheduling problem which originates from an evaporated milk processing system. The system has a semi-continuous structure. There are two continuous production stages: processing and packaging. These stages are connected by intermediate storage where materials are batch-wise standardized. The problem requires the consideration of the industry-specific characteristics of the underlying production environment. These involve traceability requirements, and time- and sequence-dependent cleaning of production units. These characteristics result in a computationally challenging scheduling problem which also requires an efficient, yet flexible modeling approach. This chapter contributes to the literature by presenting such a mathematical approach. The proposed approach decomposes and solves the overall problem in two-phases where the specifications regarding material flows are determined and a complete production schedule is developed in succession. The decomposition scheme not only simplifies the overall problem but also facilitates modeling traceability requirements by isolating material flow and scheduling decisions. The respective sub-problems concerning the two phases of the decomposition are formulated by using different modeling paradigms. The first sub-problem is formulated by using MILP. The second sub-problem, however, is formulated by employing constraint programming (CP). The approach is tested on a data set collected from a real-life evaporated milk plant and shown to be efficient. The novelty of the approach lies in coordinating the system as a whole. The majority of research contributions on scheduling food processing systems concentrate on a single production stage which is regarded as the bottleneck. However, this could only be justifiable when product and routing variety is fairly limited. This chapter mainly concentrated on the evaporated milk production process. Although this production environment is not very common within the domain of the processing industry, there are many examples of food processing systems characterized by a make-and-pack configuration. The proposed approach can be particularity appealing for such processing systems. Furthermore, it could be possible to make use of the flexibility of the current modeling scheme in order to address further elements which could be of interest in the food processing industry. For instance, the concept of chain dispersion – a measure in which production batches are spread among different customers – can easily be integrated into the proposed approach. A variety of potential directions for further research can be acknowledged. The decomposition approach taken in this chapter, in principle, leads to sub-optimal schedules. Thus, it is important to focus on approaches which can integrate the two sub-problems which are solved sequentially in the current study. Also, the computational performance of the scheduling problem comprised in the second phase can be increased by embedding more sophisticated search procedures into the CP model. Another important research direction is the consideration of possible revisions in customer orders prior to their dispatch. A possible proposal towards this issue is to integrate safety stocks and/or safety times into the scheduling approach.

Chapter 5 addressed the order acceptance problem in a food processing system where a single raw material is processed into a variety of different end products. The essence of the problem lies in the limited availability of the key raw material and the variability in yield. The customer orders for end products arrive following a stochastic process. The objective is to maximize the expected total revenue by making the optimal admission decisions for incoming orders. This chapter demonstrated that the problem can be modeled as a single resource capacity control problem, and it can be solved by means of dynamic programming (DP). However, since the structure of the optimal admission policy is found to be very complex for practice, a threshold-based heuristic policy is developed. An extensive numerical study is then conducted to compare the optimal admission policy against the new heuristic policy. The first-come-first-served policy is also included in the numerical analysis in order to reflect upon the case where no explicit admission policy is employed. This also helped to point out the cases where admission policies are critical. The results of the numerical study revealed that the heuristic policy performs very good on the overall. Also, it is observed that employing the optimal admission policy is relatively critical when the availability of the key resource is very limited, the variety of the type of customer orders is large, and the yield variability is low. The study carried out in this chapter has two main limitations. The proposed approach essentially neglects the production capacities and lead-times. Thus, the results derived in this chapter do not immediately apply to cases where production capacities are strictly limited and/or lead-times are significant. Thus, it is important to consider the problem where admission decisions must be taken in connection with both the raw material availability and the workload of the system. Nevertheless, it could be justifiable to expect that the importance of limited raw material availability diminishes in such cases since this limitation is an issue only if there is sufficient production capacity to process the raw material. Another limitation of the current study is the consideration of a single key raw material. In many production environments several raw materials are processed into end products all together. Thus, it is important to extend the current analysis to account for such systems.

Bibliography

Akkerman R (2007) Operational performance of two-stage food production systems: process interactions and capacitated storage. PhD thesis, University of Groningen

Akkerman R, Van Donk DP (2009) Analyzing scheduling in the food-processing industry: structure and tasks. Cognition, Technology & Work 11(3):215–226

Akkerman R, Van Donk DP, Gaalman G (2007) Influence of capacity-and time-constrained intermediate storage in two-stage food production systems. International Journal of Production Research 45(13):2955–2973

Akkerman R, Van Der Meer D, Van Donk DP (2010) Make to stock and mix to order: Choosing intermediate products in the food-processing industry. International Journal of Production Research 48(12):3475–3492

Amos F, Ronnqvist M, Gill G (1997) Modelling the pooling problem at the New Zealand Refining Company. Journal of the Operational Research Society 48(8):767–778

Anily S (1991) Multi-item replenishment and storage problem (MIRSP): heuristics and bounds. Operations Research 39(2):233–243

Ashayeri J, Teelen A, Selen W (1996) Computer Integrated Manufacturing in the Chemical Industry: Theory & Practice. Production and Inventory Management Journal 37(1):5257

Audet C, Brimberg J, Hansen P, Le Digabel S, Mladenovic N (2004) Pooling problem: Alternate formulations and solution methods. Management Science 50(6):761–776

Baker TE, Lasdon LS (1985) Successive linear programming at Exxon. Management Science 31(3):264–274

Baptiste P, Le Pape C, Nuijten W (2001) Constraint-based scheduling: Applying constraint programming to scheduling problems. Springer Netherlands

Belvaux G, Wolsey LA (2000) bc-prod: A specialized branch-and-cut system for lot-sizing problems. Management Science 46(5):724–738

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

Berretta R, Rodrigues LF (2004) A memetic algorithm for a multistage capacitated lot-sizing problem. International Journal of Production Economics 87(1):67–81

Bitran GR, Yanasse HH (1982) Computational complexity of the capacitated lot size problem. Management Science 28(10):1174–1186

Bollapragada R, Rao U (1999) Single-stage resource allocation and economic lot scheduling on multiple, nonidentical production lines. Management science 45(6):889–904

Brander P, Forsberg R (2005) Cyclic lot scheduling with sequence-dependent set-ups: a heuristic for disassembly processes. International Journal of Production Research 43(2):295–310

Brumelle S, Walczak D (2003) Dynamic airline revenue management with multiple semi-Markov demand. Operations Research 51:137–148

Carreno JJ (1990) Economic lot scheduling for multiple products on parallel identical processors. Management science 36(3):348–358

Castro PM, Barbosa-Povoa AP, Matos HA, Novais AQ (2004) Simple continuous-time formulation for short-term scheduling of batch and continuous processes. Industrial & Engineering Chemistry Research 43(1):105–118

Cattrysse D, Salomon M, Kuik R, Van Wassenhove LN (1993) A dual ascent and column generation heuristic for the discrete lotsizing and scheduling problem with setup times. Management Science 39(4):477–486

Caux C, David F, Pierreval H (2006) Implementation of delayed differentiation in batch process industries: a standardization problem. International Journal of Production Research 44(16):3243–3255

Chu LY, Hsu VN, Shen ZJM (2005) An economic lot-sizing problem with perishable inventory and economies of scale costs: Approximation solutions and worst case analysis. Naval Research Logistics 52(6):536–548

CIAA (2010) Confederation of the Food and Drink Industries of the EU Annual Report 2009

Clark AR, Clark SJ (2000) Rolling-horizon lot-sizing when set-up times are sequence-dependent. International Journal of Production Research 38(10):2287–2307

Cokins GM (1988) Control systems for process industries. Manufacturing Systems 6(5):79–83

Cooke DL, Rohleder TR (2006) Inventory evaluation and product slate management in large-scale continuous process industries. Journal of Operations Management 24(3):235–249

Cox JF, Blackstone JH, Spencer MS (eds) (1995) APICS Dictionary. American Production and Inventory Control Society, Falls Church, VA

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

De Matta R, Guignard M (1994a) Dynamic production scheduling for a process industry. Operations Research 42(3):492–503

De Matta R, Guignard M (1994b) Studying the effects of production loss due to setup in dynamic production scheduling. European Journal of Operational Research 72(1):62–73

De Matta R, Guignard M (1995) The performance of rolling production schedules in a process industry. IIE transactions 27(5):564–573

Dennis DR, Meredith JR (2000) An analysis of process industry production and inventory management systems. Journal of Operations Management 18(6):683–699

Dilts DM, Ramsing KD (1989) Joint lot sizing and scheduling of multiple items with sequence-dependent setup costs. Decision Sciences 20(1):120–133

Dobson G (1992) The cyclic lot scheduling problem with sequence-dependent setups. Operations Research 40(4):736–749

Dobson G, Yano CA (1994) Cyclic scheduling to minimize inventory in a batch flow line. European Journal of Operational Research 75(2):441–461

Doganis P, Sarimveis H (2008) Optimal production scheduling for the dairy industry. Annals of Operations Research 159:315–331

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

Drexl A, Haase K (1996) Sequential-analysis based randomized-regret-methods for lot-sizing and scheduling. Journal of the Operational Research Society 47(2):251–265

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

Dupuy C, Botta-Genoulaz V, Guinet A (2005) Batch dispersion model to optimise traceability in food industry. Journal of Food Engineering 70(3):333 – 339, operational Research and Food Logistics

El-Najdawi MK (1997) Multi-cyclic flow shop scheduling: an application in multi-stage, multiproduct production processes. International Journal of Production Research 35(12):3323–3332

El-Najdawi MK, Kleindorfer PR (1993) Common cycle lot-size scheduling for multi-product, multi-stage production. Management Science 39(7):872–885

Elmaghraby SE (1978) The economic lot scheduling problem (ELSP): review and extensions. Management Science 24(6):587–598

Entrup ML, Gunther HO, Van Beek P, Grunow M, Seiler T (2005) Mixed-Integer Linear Programming approaches to shelf-life-integrated planning and scheduling in yoghurt production. International Journal of Production Research 43(23):5071–5100

Ferris MC, Maravelias CT, Sundaramoorthy A (2009) Simultaneous batching and scheduling using dynamic decomposition on a grid. INFORMS Journal on Computing 21(3):398–410

Flapper SDP, Fransoo JC, Broekmeulen R, Inderfurth K (2002) Planning and control of rework in the process industries: a review. Production Planning and Control 13(1):26–34

Fleischmann B (1990) The discrete lot-sizing and scheduling problem. European Journal of Operational Research 44(3):337–348

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

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

Floudas CA, Lin X (2004) Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Computers & Chemical Engineering 28(11):2109–2129

Floudas CA, Lin X (2005) Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. Annals of Operations Research 139(1):131–162

Forza C, Salvador F, Trentin A (2008) Form postponement effects on operational performance: A typological theory. International Journal of Operations & Production Management 28(11):1067–1094

Franca PM, Armentano VA, Berretta RE, Clark AR (1997) A heuristic method for lot-sizing in multi-stage systems. Computers & Operations Research 24(9):861–874

Fransoo JC (1993) Production control and demand management in capacitated flow process industries. PhD thesis, Technische Universiteit Eindhoven

Fransoo JC, Rutten W (1994) A typology of production control situations in process industries. International Journal of Operations and Production Management 14:47–47

Gaglioppa F, Miller LA, Benjaafar S (2008) Multitask and multistage production planning and scheduling for process industries. Operations Research 56(4):1010–1025

Gallego G, Queyranne M, Simchi-Levi D (1996) Single resource multi-item inventory systems. Operations Research 44(4):580–595

Galvin TM (1987) Economic lot scheduling problem with sequence-dependent setup costs. Production and Inventory Management Journal 28(1):96–105

Gimenez DM, Henning GP, Maravelias CT (2009) A novel network-based continuous-time representation for process scheduling: Part i. main concepts and mathematical formulation. Computers & Chemical Engineering 33(9):1511 – 1528

Glen JJ (1980) A mathematical programming approach to beef feedlot optimization. Management Science 26(5):524–535

Goyal SK (1994) A note on 'effect of production cost on shelf life'. International Journal of Production Research 32(9):2243

Grosfeld-Nir A, Gerchak Y (2004) Multiple lotsizing in production to order with random yields: review of recent advances. Annals of Operations Research 126(1):43–69

Gunasekaran A (1998) Concurrent engineering: a competitive strategy for process industries. Journal of the Operational Research Society 49(7):758–765

Gupta D, Magnusson T (2005) The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times. Computers & Operations Research 32(4):727–747

Gupta S, Karimi IA (2003) Scheduling a two-stage multiproduct process with limited product shelf life in intermediate storage. Industrial & Engineering Chemistry Research 42(3):490–508

Ha JK, Chang HK, Lee ES, Lee IB, Lee BS, Yi GB (2000) Intermediate storage tank operation strategies in the production scheduling of multi-product batch processes. Computers & Chemical Engineering 24:1633 – 1640

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

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

Hanssmann F (1962) Operations research in production and inventory control. John Wiley & Sons, New York City

Harjunkoski I, Grossmann IE (2001) A decomposition approach for the scheduling of a steel plant production. Computers & Chemical Engineering 25(11-12):1647–1660

Hsu VN (2000) Dynamic economic lot size model with perishable inventory. Management Science 46(8):1159–1169

Hsu WL (1983) On the general feasibility test of scheduling lot sizes for several products on one machine. Management Science 29(1):93–105

Ierapetritou MG, Floudas CA (1998) Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. Industrial & Engineering Chemistry Research 37:4341–4359

ILOG (2008) OPL Studio 6.0. User's Manual

Jain V, Grossmann I (2001) Algorithms for hybrid MILP/CP models for a class of optimization problems. INFORMS Journal on Computing 13(4):258–276

Jain V, Grossmann IE (2000) A Disjunctive Model for Scheduling in a Manufacturing and Packing Facility with Intermediate Storage. Optimization and Engineering 1:215–231

Janak SL, Lin X, Floudas CA (2004) Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: Resource constraints and mixed storage policies. Industrial & Engineering Chemistry Research 43(10):2516–2533

Jans R, Degraeve Z (2004) An industrial extension of the discrete lot-sizing and scheduling problem. IIE Transactions 36(1):47–58

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

Jenabi M, Fatemi Ghomi SMT, Torabi SA, Karimi B (2007) Two hybrid meta-heuristics for the finite horizon ELSP in flexible flow lines with unrelated parallel machines. Applied Mathematics and Computation 186(1):230–245

Jones PC, Inman RP (1989) When is the Economic Lot Scheduling Problem Easy? IIE Transactions 21(1):11–20

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

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

Karmarkar US, Rajaram K (2001) Grade selection and blending to optimize cost and quality. Operations Research 49(2):271–280

Karmarkar US, Kekre S, Kekre S (1987) The dynamic lot-sizing problem with startup and reservation costs. Operations Research 35(3):389–398

Katok E, Lewis HS, Harrison TP (1998) Lot sizing in general assembly systems with setup costs, setup times, and multiple constrained resources. Management Science 44(6):859–877

Kelly JD, Zyngier D (2007) An improved MILP modeling of sequence-dependent switchovers for discrete-time scheduling problems. Industrial & Engineering Chemistry Research 46(14):4964–4973

Kilic OA, Akkerman R, Grunow M, Van Donk DP (2009) Modeling intermediate product selection under production and storage capacity limitations in food processing. In: Proceedings of the International Conference on Industrial Engineering and Engineering Management, pp 1077 – 1081

Kilic OA, Van Donk DP, Wijngaard J, Tarim SA (2010) Order acceptance in food processing systems with random raw material requirements. OR Spectrum 32(4):905–925

Kilic OA, Van Donk DP, Wijngaard J (2011) A discrete time formulation for batch processes with storage capacity and storage time limitations. Computers & Chemical Engineering 35(4):622–629

Kleywegt AJ, Papastavrou JD (2001) The dynamic and stochastic knapsack problem with random sized items. Operations Research 49:26–41

Kondili E, Pantelides CC, Sargent RWH (1993a) A general algorithm for short-term scheduling of batch-operations. I. MILP formulation. Computers & Chemical Engineering 17:211 – 227

Kondili E, Shah N, Pantelides CC (1993b) Production planning for the rational use of energy in multiproduct continuous plants. Computers & Chemical Engineering 17:123–128

Kopanos G, Puigjaner L, Georgiadis M (2010) Optimal production scheduling and lot-sizing in dairy plants: the yogurt production line. Industrial & Engineering Chemistry Research 49(2):701–718

Laguna M (1999) A heuristic for production scheduling and inventory control in the presence of sequence-dependent setup times. IIE transactions 31(2):125–134

Lasdon LS, Terjung RC (1971) An efficient algorithm for multi-item scheduling. Operations Research 19(4):946–969

Lee CY (1989) A Solution to the Multiple Set-Up Problem with Dynamic Demand. IIE transactions 21(3):266–270

Lee TC, Hersh M (1993) A model for dynamic airline seat inventory control with multiple seat bookings. Transportation Science 27:252–265

Lippman SA (1969) Optimal inventory policy with multiple set-up costs. Management Science 16(1):118–138

Lopez MAN, Kingsman BG (1991) The economic lot scheduling problem: theory and practice. International Journal of Production Economics 23(1-3):147 – 164

Maravelias CT, Grossmann IE (2003) Minimization of the makespan with a discrete-time state-task network formulation. Industrial & Engineering Chemistry Research 42(24):6252 – 6257

Maravelias CT, Grossmann IE (2006) On the relation of continuous- and discrete-time state-task network formulations. AIChE Journal 52:843 – 849

Marinelli F, Nenni ME, Sforza A (2007) Capacitated lot sizing and scheduling with parallel machines and shared buffers: A case study in a packaging company. Annals of Operations Research 150(1):177–192

McIntosh RI, Matthews J, Mullineux G, Medland AJ (2010) Late customisation: issues of mass customisation in the food industry. International Journal of Production Research 48(6):1557–1574

Mendez CA, Cerda J (2002) An MILP-based approach to the short-term scheduling of make-and-pack continuous production plants. OR Spectrum 24:403–429

Mendez CA, Cerda J, Grossmann IE, Harjunkoski I, Fahl M (2006) State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers & Chemical Engineering 30:913 – 946

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

Mockus L, Reklaitis GV (1997) Mathematical programming formulation for scheduling of batch operations based on nonuniform time discretization. Computers & Chemical Engineering 21(10):1147–1156

Murthy DNP, Ma L (1996) Material planning with uncertain product quality. Production Planning & Control 7(6):566–576

Nahmias S (1982) Perishable inventory theory: A review. Operations Research 30(4):680–708

Neumann K, Schwindt C, Trautmann N (2002) Advanced production scheduling for batch plants in process industries. OR Spectrum 24(3):251–279

Ouenniche J, Bertrand JWM (2001) The finite horizon economic lot sizing problem in job shops::: the multiple cycle approach. International Journal of Production Economics 74(1-3):49–61

Ouenniche J, Boctor F (1998) Sequencing, lot sizing and scheduling of several products in job shops: the common cycle approach. International journal of production research 36(4):1125–1140

Ouenniche J, Boctor FF, Martel A (1999) The impact of sequencing decisions on multi-item lot sizing and scheduling in flow shops. International Journal of Production Research 37(10):2253–2270

Ozdamar L, Barbarosoglu G (1999) Hybrid heuristics for the multi-stage capacitated lot sizing and loading problem. Journal of the Operational Research Society 50(8):810–825

Ozdamar L, Barbarosoglu G (2000) An integrated Lagrangean relaxation-simulated annealing approach to the multi-level multi-item capacitated lot sizing problem. International Journal of Production Economics 68(3):319–331

Ozdamar L, Birbil SI (1998) 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

Papageorgiou LG, Pantelides CC (1996a) Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 1. Mathematical formulation. Industrial & Engineering Chemistry Research 35(2):488–509

Papageorgiou LG, Pantelides CC (1996b) Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 2. A mathematical decomposition approach. Industrial & Engineering Chemistry Research 35(2):510–529

Papastavrou JD, Rajagopalan S, Kleywegt AJ (1996) The dynamic and stochastic knapsack problem with deadlines. Management Science 42:1706–1718

Pesenti R, Ukovich W (2003) Economic lot scheduling on multiple production lines with resource constraints. International Journal of Production Economics 81:469–481

Pochet Y, Wolsey L (1993) Lot-sizing with constant batches: Formulation and valid inequalities. Mathematics of Operations Research 18(4):767–785

Potts CN, Kovalyov MY (2000) Scheduling with batching: a review. European Journal of Operational Research 120(2):228–249

Raza AS, Akgunduz A (2008) A comparative study of heuristic algorithms on Economic Lot Scheduling Problem. Computers & Industrial Engineering 55(1):94–109

Rice JW, Norback JP (1987) Process industries production planning using matrix data structures. Production and Inventory Management Journal 28(2):15–23

Rippin DWT (1991) Batch process planning. Chemical Engineering 98(5):100–7

Rogers J (1958) A computational approach to the economic lot scheduling problem. Management Science 4(3):264–291

Rong A, Grunow M (2010) A methodology for controlling dispersion in food production and distribution. OR Spectrum 32:957–978

Rutten W (1993) Hierarchical mathematical programming for operational planning in a process industry. European Journal of Operational Research 64(3):363–369

Sahinidis NV, Grossmann IE (1991a) MINLP model for cyclic multiproduct scheduling on continuous parallel lines. Computers & Chemical Engineering 15(2):85–103

Sahinidis NV, Grossmann IE (1991b) Reformulation of multiperiod MILP models for planning and scheduling of chemical processes. Computers & Chemical Engineering 15:255–272

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

Salomon M, Kroon LG, Kuik R, Van Wassenhove LN (1991) Some extensions of the discrete lotsizing and scheduling problem. Management Science 37(7):801–812

Salomon M, Solomon MM, Van Wassenhove LN, Dumas Y, Dauzere-Peres S (1997) Solving the discrete lotsizing and scheduling problem with sequence dependent set-up costs and set-up times using the travelling salesman problem with time windows. European Journal of Operational Research 100(3):494–513

Sarker BR, Babu PS (1993) Effect of production cost on shelf life. International Journal of Production Research 31(8):1865–1872

Schilling G, Pantelides CC (1996) A simple continuous-time process scheduling formulation and a novel solution algorithm. Computers & Chemical Engineering 20:1221–1226

Schultmann F, Fröhling M, Rentz O (2006) Fuzzy approach for production planning and detailed scheduling in paints manufacturing. International Journal of Production Research 44(8):1589–1612

Schuster EW, Allen SJ (1998) Raw Material Management at Welch's, Inc. Interfaces 28(5):13–24

Schwindt C, Trautmann N (2000) Batch scheduling in process industries: an application of resource–constrained project scheduling. OR Spectrum 22(4):501–524

Shah N, Pantelides CC, Sargent RWH (1993) A general algorithm for short-term scheduling of batch operations–II. Computational issues. Computers & Chemical Engineering 17(2):229–244

Silver EA (1989) Shelf life considerations in a family production context. International Journal of Production Research 27(12):2021–2026

Silver EA (1995) Dealing with a shelf life constraint in cyclic scheduling by adjusting both cycle time and production rate. International Journal of Production Research 33(3):623–629

Silver EA, Pyke DF, Peterson R (1998) Inventory management and production planning and scheduling, 3rd edn. John Wiley & Sons, New York, NY

Soman CA, Van Donk DP, Gaalman G (2004a) Combined make-to-order and make-to-stock in a food production system. International Journal of Production Economics 90(2):223–235

Soman CA, Van Donk DP, Gaalman GJC (2004b) A basic period approach to the economic lot scheduling problem with shelf life considerations. International Journal of Production Research 42(8):1677–1689

Somsen D, Capelle A (2002) Introduction to production yield analysis–a new tool for improvement of raw material yield. Trends in Food Science & Technology 13:136 – 145

Steuer RE (1984) Sausage blending using multiple objective linear programming. Management Science 30(11):1376–1384

Stigler GJ (1945) The cost of subsistance. Journal of Farm Economics 27:303–14

Suerie C (2006) Modeling of period overlapping setup times. European Journal of Operational Research 174(2):874–886

Sundaramoorthy A, Maravelias CT (2008) Modeling of storage in batching and scheduling of multistage processes. Industrial & Engineering Chemistry Research 47(17):6648–6660

Susarla N, Li J, Karimi IA (2010) A novel approach to scheduling multipurpose batch plants using unit-slots. AIChE Journal 56(7):1859–1879

Talluri KT, Van Ryzin GJ (2004) The theory and practice of revenue management. Kluwer Academic Publishers, Boston, MA.

Tamayo S, Monteiro T, Sauer N (2009) Deliveries optimization by exploiting production traceability information. Engineering Applications of Artificial Intelligence 22(4-5):557 – 568

Taube-Netto M (1996) Integrated planning for poultry production at Sadia. Interfaces 26(1):38–53

Taylor SG, Seward SM, Bolander SF (1981) Why the process industries are different. Production and Inventory Management Journal 22(4):9–24

Tempelmeier H, Derstroff M (1996) A Lagrangean-based heuristic for dynamic multilevel multiitem constrained lotsizing with setup times. Management Science 42(5):738–757

Thakur M, Wang L, Hurburgh CR (2010) A multi-objective optimization approach to balancing cost and traceability in bulk grain handling. Journal of Food Engineering 101(2):193 – 200

Torabi SA, Karimi B, Fatemi Ghomi SMT (2005) The common cycle economic lot scheduling in flexible job shops: The finite horizon case. International Journal of Production Economics 97(1):52–65

Van Donk DP (2000) Customer-driven manufacturing in the food processing industry. British Food Journal 102:739–747

Van Donk DP (2001) Make to stock or make to order: The decoupling point in the food processing industries. International Journal of Production Economics 69(3):297–306

Van Donk DP, Fransoo JC (2006) Operations management research in process industries. Journal of Operations Management 24(3):211–214

Van Hentenryck P (1999) The OPL Optimization Programming Language. MIT Press, Cambridge, Massachusetts

Van Hoek RI (1999) Postponement and the reconfiguration challenge for food supply chains. Supply Chain Management: An International Journal 4(1):18–34

Van Kampen T, Van Donk DP, Van der Zee DJ (2010) Safety stock or safety lead time: coping with unreliability in demand and supply. International Journal of Production Research 48(24):7463–7481

Van Slyke R, Young Y (2000) Finite horizon stochastic knapsacks with applications to yield management. Operations Research 48:155–172

Van Vyve M, Ortega F (2004) Lot-sizing with fixed charges on stocks: the convex hull. Discrete Optimization 1(2):189–203

Van Wezel W, Van Donk DP, Gaalman G (2006) The planning flexibility bottleneck in food processing industries. Journal of Operations Management 24(3):287 – 300

Vanderbeck F (1998) Lot-Sizing with Start-Up Times. Management Science 44(10):1409

Venkatesh S, Swaminathan JM (2003) Managing product variety through postponement: Concept and applications. In: Harrison T, Lee H, Neale J (eds) The Practice of Supply Chain Management: Where Theory and Applications Converge, Kluwer Publishers, Boston, MA, pp 139–155

Viswanathan S (1995) A note on Effect of production cost on shelf life. International Journal of Production Research 33(12):3485–3486

Viswanathan S, Goyal SK (1997) Optimal cycle time and production rate in a family production context with shelf life considerations. International Journal of Production Research 35(6):1703–1712

Viswanathan S, Goyal SK (2000) Incorporating planned backorders in a family production context with shelf-life considerations. International Journal of Production Research 38(4):829–836

Viswanathan S, Goyal SK (2002) On “manufacturing batch size and ordering policy for products with shelf lives”. International Journal of Production Research 40(8):1965 – 1970

Wagner BJ, Davis DJ (2002) A search heuristic for the sequence-dependent economic lot scheduling problem. European Journal of Operational Research 141(1):133–146

Walstra P, Wouters JTM, Geurts TJ (2006) Dairy Science and Technology Handbook, 2nd edn. Taylor and Francis, Boca Raton, FL

Wang X, Li D, O’Brien C (2009) Optimisation of traceability and operations planning: an integrated model for perishable food production. International Journal of Production Research 47(11):2865–2886

Wang X, Li D, O’Brien C, Li Y (2010) A production planning model to reduce risk and improve operations management. International Journal of Production Economics 124(2):463 – 474

Williams HP, Redwood AC (1974) A structured linear programming model in the food industry. Operational Research Quarterly 25(4):517–527

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

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

Samenvatting (Summary in Dutch)

De procesindustrie heeft te maken gehad met toenemende logistieke eisen, een groeiende verscheidenheid in producten en toenemende concurrentie. De combinatie van deze trends heeft in deze bedrijfstak gezorgd voor een sterke focus op de toepassing van efficiënte planning en scheduling. Het doel van dit proefschrift is om bij te dragen aan de kennis over planning en scheduling in de procesindustrie waarbij in het bijzonder rekening wordt gehouden met de industriespecifieke kenmerken van productiesystemen.

De procesindustrie kent specifieke product- en proceskenmerken die het beheersen van de productie aanzienlijk beïnvloeden. Het gaat hierbij om overwegingen met betrekking tot de beschikbaarheid en opbrengst van grondstoffen, flexibele productrecepten, bederfelijke materialen, traceerbaarheidsvereisten, opslagcapaciteit beperkingen, en productie omstellingen. Ieder van deze kenmerken is van praktisch belang. Echter, in veel productieomgevingen komen een aantal van deze kenmerken tegelijkertijd voor. Daarom moet er in de planning en scheduling vaak rekening gehouden worden met een combinatie van een aantal industriespecifieke kenmerken en tegelijkertijd met de interacties tussen deze kenmerken. De literatuur biedt een verscheidenheid aan modellen en methoden die deze kenmerken afzonderlijk in ogenschouw nemen. Toch lijkt het erop dat de relatie tussen de kenmerken onvoldoende aan bod is gekomen. Het onderzoek in dit proefschrift kan worden gezien als een poging om dit hiaat in de literatuur te vullen.

Hoofdstuk 1 introduceert de relevante product- en proceskenmerken en bespreekt de scheduling methoden die gebruikt worden in de procesindustrie. Dit hoofdstuk geeft een kritische beschouwing van de literatuur en onderzoekt de mate waarin de industriespecifieke kenmerken worden meegenomen in planning en scheduling in de procesindustrie. Dit overzicht vormt de basis van de onderzoeksagenda. De rest van het proefschrift is georganiseerd als een collectie van wetenschappelijke artikelen rond het hoofdthema van het proefschrift. Deze artikelen zijn gewijd aan afzonderlijke en praktisch-relevante problemen afkomstig uit specifieke productieomgevingen. Elk individueel paper ontwerpt een wiskundig model en ontwikkelt vervolgens oplossingsmethoden voor het probleem op basis van goed gefundeerde optimalisatiemethoden.

Hoofdstuk 2 behandelt een schedulingsprobleem in twee fase productiesystemen van levensmiddelen met productie- en opslagcapaciteit beperkingen. De essentie van dit probleem betreft het gebruik van flexibele productrecepten. In wezen is het probleem een uitbreiding van het production lot scheduling probleem met geïntegreerde ontwerpbeslissingen. Een uitgebreid Mixed Integer Linear Programming (MILP) model is ontwikkeld dat de beslissingen ten aanzen van het productieschema alsmede de specificaties van halffabricaten en eindproducten integreert, met als doel de totale operationele kosten te minimaliseren. Het model is vervolgens toegepast op een dataset die betrekking heeft op een werkelijk probleem. De resultaten uit de numerieke studie zijn gebruikt om beter inzicht te krijgen in de dynamiek van het probleem. Aangetoond wordt dat de kostenparameters en capaciteitsbeperkingen het productieschema en de selectie van productrecepten significant beïnvloeden. De studie demonstreert dat de vaste productie omstelkosten en voorraadkosten beinvloed worden door het veranderen van het aantal tussenproducten en de recepten van de eindproduct maar ook door het productieschema. Ook de afwegingen tussen de capaciteitsbeperkingen en operationele kosten zijn onderzocht. Aangetoond wordt dat er een wisselwerking is tussen de beperkingen van de productiecapaciteit en de beperkingen van de opslagcapaciteit. Dus de vraag of een bepaald type capaciteitsbeperking bindend is hangt af van zijn relatieve grootte.

Hoofdstuk 3 concentreert zich op het gedetailleerde kortetermijn schedulingsprobleem in multiproduct/multifunctionele batchprocessen. In dit hoofdstuk wordt het conventionele discrete tijd MILP model voor het plannen van batch-processen uitgebreid door het introduceren van opslagcapaciteit- en opslagtijdbeperkingen. Deze beperkingen komen veel voor in bedrijfstakken die te maken hebben met bederfelijke halffabricaten en eindproducten. Er wordt een wiskundig model voor dit probleem ontwikkeld en er wordt aangetoond dat dit model geschikt is voor verschillende opslagconfiguraties: zowel enkele als meervoudige en productspecifieke als multifunctionele opslagtanks. Door middel van een numerieke studie wordt geïllustreerd hoe opslagcapaciteit- en de opslagtijdbeperkingen een significant effect kunnen hebben op productie- en opslagactiviteiten en hoe de kosten van dergelijke batch-gewijs verwerking systemen significant kunnen worden verminderd. Daarnaast wordt aangetoond dat deze effecten tot op zekere hoogte kunnen worden verminderd door middel van het gebruik van multifunctionele opslagtanks.

Hoofdstuk 4 onderzoekt een schedulingsprobleem dat afkomstig is uit een bedrijf dat gespecialiseerd is in de productie van gecondenseerde melk. Het productieproces heeft een semicontinue structuur. Er zijn twee continue productiestappen: verwerking en verpakking. Deze stappen zijn verbonden door een tussentijdse opslag waar materialen batch-gewijs worden gestandaardiseerd. De productieomgeving heeft een aantal industriespecifieke kenmerken zoals traceerbaarheidsvereisten en tijd- en volgordeafhankelijke reiniging van productie-installaties. Het hoofdstuk presenteert een wiskundige benadering voor dit probleem bestaande uit twee fasen. In deze fasen wordt achtereenvolgens de specificaties ten aanzien van de materiaalstromen bepaald en een volledig productieschema gebouwd. Deze decompositie vereenvoudigt niet alleen het algemene probleem maar vergemakkelijkt ook het modelleren van traceerbaarheidsvoorschriften door het isoleren van de materiaalstroom en de schedulingsbeslissingen. De sub-problemen behorende bij de twee fasen van de decompositie worden gemodelleerd met behulp van verschillende modelleringsparadigma's. Het eerste subprobleem is gemodelleerd door gebruik te maken van MILP. Het tweede subprobleem is gemodelleerd door middel van Constraint Programming. De aanpak is getest op een dataset afkomstig van deze producent van gecondenseerde melk en de testen tonen aan dat de gebruikte aanpak efficiënt is.

Hoofdstuk 5 richt zich op het orderacceptatie probleem in een productieproces van een voedingsmiddelenproducent waarin een enkele grondstof verwerkt wordt tot een verscheidenheid aan verschillende eindproducten. De essentie van het probleem ligt in de beperkte beschikbaarheid van de belangrijkste grondstof en de variabiliteit in opbrengst. De doelstelling van het probleem is om de verwachte totale opbrengst te maximaliseren door het maken van optimale acceptatiebeslissingen voor binnenkomende orders. Aangetoond wordt dat het probleem kan worden gemodelleerd als een capaciteitsbeheersingsprobleem met een enkele productie-eenheid. Echter, aangezien de structuur van het optimale acceptatiebeleid zeer complex blijkt te zijn voor de praktijk is een eenvoudig heuristische beslisregel ontwikkeld die op basis van een bepaalde drempelwaarde orders accepteert. Een uitgebreide numerieke studie is uitgevoerd om het optimale acceptatiebeleid te vergelijken met de heuristische regels. De resultaten van de numerieke studie tonen aan dat de heuristische beslisregel in het algemeen zeer goed presteert. Ook constateren we dat het gebruik van het optimale toelatingsbeleid cruciaal is wanneer de beschikbaarheid van de belangrijkste productie-eenheid zeer beperkt is, de verscheidenheid van het type orders groot is, en de variabiliteit van de opbrengst laag is.

Hoofdstuk 6 presenteert tot slot de onderzoeksresultaten van het proefschrift, geeft een reflectie op het uitgevoerde onderzoek en formuleert enkele suggesties voor toekomstig onderzoek.