552 LECTURE NOTES IN ECONOMICS AND MATHEMATICAL SYSTEMS

Christopher Suerie

Time Continuity in Discrete Time Models

New Approaches for Production Planning in Process Industries

Springer

Lecture Notes in Economics and Mathematical Systems 552

Christopher Suerie

Time Continuity in Discrete Time Models

New Approaches for Production Planning in Process Industries

Springer

Author

Christopher Suerie
Department of Production & Supply Chain Management
Technische Universität Darmstadt
Hochschulstrasse 1
64289 Darmstadt
Germany
E-mail: suerie@bwl.tu-darmstadt.de

Dissertation an der Technischen Universität Darmstadt (D17)

Library of Congress Control Number: 2005921140

ISSN 0075-8442
ISBN 3-540-24521-9 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera ready by author
Cover design: *Erich Kirchner*, Heidelberg

Printed on acid-free paper 42/3130Di 5 4 3 2 1 0

Foreword

In recent years great efforts have been made in industry to reduce complexity of production processes and to lower setup times and setup cost. Still, we have observed numerous production facilities where lot-sizing continues to play a major role. Also, the issue of lot-sizing spans a much larger area than merely minimizing the sum of setup and holding costs as it also provides the clue for a better utilization of resources. For example, the author is aware of a case where improved lot-sizing and scheduling increased output by more than 20%!

Still the question remains which lot-sizing model to choose. There is a vast number of lot-sizing models in the literature either based on a discrete time axis or on a continuous times axis. While the former is easier to solve in general aggregation of time often results in missing "optimal" solutions or even feasible solutions (although these might exist). Continuous time models, despite being able to capture more details, often are complex non-linear models resulting in prohibitive computational efforts for its solution.

This was the situation when Christopher Suerie started his PhD project. In the course of the project he came up with a number of excellent ideas to improve modeling capabilities of discrete time model formulations. In the end he has been able to claim that now mixed integer linear model formulations for the *capacitated lot-sizing problem with linked lot sizes* (CLSPL) as well as the *proportional lot-sizing and scheduling problem* (PLSP) exist capturing details that make continuous time model formulations unnecessary. To be more precise, Christopher Suerie has shown how to effectively model restrictions on period overlapping lot sizes (campaigns), namely

- minimal and maximal production amounts,
- minimal resource utilizations throughout campaign production and
- production amounts that are integer multiples of a given batch size.

Furthermore, he has developed a model formulation that mimics period overlapping setup times. He also demonstrates that all his proposals are solvable by state-of-the-art Mixed Integer Programming solvers with rather modest computational efforts – thus making it most appealing for applications in industrial practice.

In the end this PhD thesis not only contributes to a number of single issues that have been treated incorrectly or ineffectively in the literature but provides a comprehensive, unifying modeling framework for single stage lot-sizing and scheduling problems directly applicable in the process industries. It is an excellent piece of research with great potentials for successful applications and worth reading from the first line until the very end.

Hamburg, January 2005 Hartmut Stadtler

Preface

This dissertation is the result of a four-year research effort conducted at the department of Production and Supply Chain Management at Darmstadt University of Technology.

In the beginning of this research we set out to include special characteristics observed in the process industries into mathematical models and algorithms for mid-term production planning. However, after some time I came up looking at a bigger picture. Having analyzed the representation defects of lot-sizing models based on a discrete time scale, I was wondering if it was possible to overcome these defects within this handy time structure. The outcome are mathematical programming model formulations and a temporal decomposition heuristic to model and solve production planning problems of the process industries suffering from the representation defect imposed by time discretization.

This work would not have been made possible without the backing of numerous supporters. First of all, I am deeply indebted to my advisor Professor Dr. Hartmut Stadtler. He not only challenged my efforts by consistently raising new questions, but also encouraged and promoted me as best as possible. As an expert in lot-sizing he was able to provide lots of valuable input in all stages of this dissertation project. Furthermore, I would like to thank Professor Dr. Wolfgang Domschke for his willingness to serve as co-adviser and second referee of this dissertation. His broad expertise and interest in optimization and operations research proved to be a priceless source of information.

Next to my academic advisors, I would like to acknowledge the contribution of my colleagues, who always provided a fruitful working environment and served as interested discussion partners at numerous occasions. Especially, I would like to thank Dr. Jens Rohde for shouldering much of the day-to-day work when I was busy doing research. Moreover, Dr. Gregor Dudek and Martin Albrecht eagerly listened to my new ideas and proofread parts or all of the manuscript. Last but not least I would like to thank Bernd Wagner for providing hardware and software as well as his expertise to conduct some of the computational tests.

However, there is a life besides academia. Nothing of the above would have been materialized without the support of my family. I would like to thank my parents for providing me with the education that enabled me to write this dissertation. Furthermore, I am happy that there is my wife Martina. Although she had to miss myself too often, she backed my efforts during highs and lows. On top of that, she proofread the manuscript several times.

Thanks to all of you.

Darmstadt, January 2005 Christopher Suerie

Table of Contents

1 Introduction

1.1 Motivation

Different modeling paradigms often collide at the interface of short-term, operational production planning and mid-term production planning. Mid-term plans are most often based on a discrete time scale made of weekly or monthly buckets without too much detail. On the other hand, short-term operational planning needs a lot more detail and therefore comprises time buckets with the size of days or shifts – or even better – is not attached to a fixed grid of time buckets, that is a continuous time scale.

Both paradigms have their legitimacy in their respective settings. For mid-term planning it is sufficient to know, that e.g. products A, B and C will be produced in the quantities 50, 80 and 30 units in week 17. On the other hand, it is important to know that the setup change from product A to B for the stamping machine needs to take place on e.g. Tuesday between 2.30 p.m. and 5 p.m., because setup personnel has to be scheduled for this event.

Models and algorithms for both, production planning on a discrete time scale and for production planning on a continuous time scale, are known in large numbers. A missing link and the focus of this thesis will be the representation of arbitrary (continuous) plans on a discrete time scale.

From a theoretical point of view this idea is very appealing, as it would allow to combine short-term and mid-term planning into one modeling approach. If a telescopic time scale with shorter time buckets at the beginning, to capture the detail necessary for short-term production planning, and bigger time buckets towards the end of the planning horizon is used, both planning steps can be accomplished with only one model. As a consequence, the structural differences which often complicate communication at the interface of short-term and mid-term production planning are reduced.

Anyhow, not a global model that solves all kinds of production planning problems will be presented here, but rather several important building blocks, primarily intended for mid-term production planning and thus bucket-oriented will be introduced. These building blocks may be used as different extensions to standard lot-sizing models. They are motivated by practical production planning problems. Moreover, built together into one model, it will be possible to represent arbitrary continuous production plans in a bucket-oriented setting.

The application of these planning models, which first comes into mind, is process industries. Furthermore, also discrete production environments might be eligible for use of at least some of the building blocks that will be presented. This stems not only from the analogy between discrete production and process industries,[1] but can also be seen from the case descriptions which follow in section 1.3.

1.2 Some Definitions

In different industries different terms are used – from a planning point of view – in the same or similar context. Here the focus will be on the terms "lot-sizing" and "campaign planning" first, which are in fact terms originating from totally different sources.

The term "lot-sizing" has its roots in a discrete production environment. Lot-sizing is the arrangement of demands for the same product in different periods to a single production order ("lot").[2] This means, that customer orders (or anonymous demand) with different due dates for a certain product are combined to form a production order. The reasoning is, that each production order is usually associated with a certain fixed cost (setup cost). If customer orders were produced as demanded ("lot-for-lot"-production), this would strongly affect costs. Furthermore, it would affect capacity, because setups will generally consume also a fixed amount of capacity. To avoid that, customer orders are combined. The result of lot-sizing is a production plan, which shows when to produce (e.g., in week 13) and how much (e.g., 100 units).

The term "campaign planning" on the other hand is typically used in the process industries. There, two variants of campaigns have to be distinguished: single-product campaigns and multi-product campaigns.[3] In analogy to lot-sizing a single-product-campaign can be defined as a production order, which comprises several customer orders (or anonymous demand) that share an unique setup state. With respect to the production environment there may be several specialties or additional constraints. The most important one is, that the production order may be made of several batches, with a batch defined as a combination of a production amount and a certain task.[4] The batch size is often fixed and determined by the size of the production resource (e.g., a tank). Anyhow, in general – at least at this level of abstraction – there is not a big difference between a lot in lot-sizing and a single-product campaign in campaign planning.

Multi-product campaigns do not fit into the lot-sizing scheme as easily. Here, a campaign consists of several products requiring different setup states, but campaigns are built such, that setup operations within a campaign require much less effort (cost and/or time) than setup operations between campaigns.[5] An analogy in lot-sizing is the grouping of products into families such that only minor setups are

---

1 Cf. Voß and Witt (2003) pp. 75-81.

2 E.g., Chase et al. (1998) pp. 648-649, Günther and Tempelmeier (1997) p. 182, Gutenberg (1983) p. 201 and Silver et al. (1998) p. 198.

3 Cf. Blömer (1999) p. 16 and Overfeld (1990) pp. 87-88.

4 Cf. Schwindt and Trautmann (2000) p. 502.

5 Cf. Overfeld (1990) pp. 87-88.

necessary between members of a product family and major setups are necessary between families.[6]

Moreover, a term used in discrete production as well as in the process industries is "batching". Unfortunately the meaning is different in both contexts. In the process industries a batch is defined as a combination of a production amount and a certain task. If the batch size is not determined by the production resources, the decision on batch sizes is called batching here (first step).[7] In a second step, batches requiring the same resource configuration (setup) are combined to form a campaign. The reason why production planning in the process industries is often in batches and the batches are not put together to form a bigger batch for planning purposes is, that resources or tanks often limit certain production tasks.[8] In discrete production however, the second step is referred to as batching. Here, the combination of production orders belonging to the same order family is called batching.[9] For a more extensive discussion we refer to Voß and Witt (2003), who discuss the different meanings of batching in discrete production and process industries as well as find and define analogous terms in these two fields.[10]

In the remainder of this thesis we will stick to the following nomenclature:

- A lot (or lot size) is the production amount of a production order which is produced in one production run without changing the setup state.
- As there are only settings in the scope of this thesis which require the planning of single-product campaigns, the term "campaign" can be used as a synonym to the term "lot".

Anyhow, when literature from the different fields is discussed in chapters 2 and 4, the term "lot" will be used, if the lot size is produced within a period (time bucket), whereas the term "campaign" will be used, if the lot extends over several periods. But this is only not to confuse readers familiar with only one field and one should keep in mind, that generally – at this level of abstraction – there is no big difference between lot-sizing and campaign planning, apart from the latter requiring some side constraints.

Furthermore, we will refrain from using the term "batching" to avoid confusion of the reader, because this term – as mentioned above – has very different meanings in discrete production and in the process industries.

1.3 Case Studies

The production planning models studied in this thesis are not only interesting from a theoretical point of view, but also relevant from a practitioner's perspective, as the following case descriptions illustrate.

---

6 Cf. Potts and van Wassenhove (1992) p. 397.
7 Cf. Trautmann (2001) p. 5.
8 Cf. Voß and Witt (2003) pp. 76-77, 81.
9 Cf. Potts and Kovalyov (2000) pp. 228, 231 and Voß and Witt (2003) pp. 78, 81.
10 Cf. Voß and Witt (2003) pp. 75-81.

- *Napkin production*
  The production process consists of three stages. At the first stage paper is produced in a continuous process. The second processing step – and in this case the bottleneck – is the conversion of paper into napkins. Here, an emblem or design is printed on the paper, which is then folded into shape. The folding operation involves a difficult setup step, which takes up to 36 hours. At the third stage the folded napkins are wrapped and packaged.[11]
  Although production plans in this case assume a period length of approximately one month, setup operations consume a substantial portion of time (5 % of capacity) and therefore need to be accounted for as accurately as possible. Otherwise production capacity may be lost or orders that should have been taken are declined.
- *Self-adhesive laminate*
  Self-adhesive laminates comprise of two layers. The top layer is formed by the face material made of paper or a synthetic, which is usually coated. On the back-side of the top layer an adhesive is applied. The bottom layer is mostly made of paper, which is silicon coated for easier release of the top layer.[12]
  In this case the bottleneck to be planned for is the coating of the face material. The planning horizon is three weeks and the varnish/paste coater is utilized five days per week and 24 hours per day. The period length is one day and the setup time is about 5 % of capacity. It is not only important to account for setup times correctly because of the tight capacity situation, but also because setups waste energy and raw materials, which go into scrap.[13]
- *Production in a chemical plant I*
  A chemical plant is considered in this case. Here, a reactor has to be planned for. This reactor can operate in different modes, producing exactly one distinct product in each mode. Changeovers are not only very costly, but also consume a considerable portion of available capacity. The planning horizon comprises one to three years with monthly buckets. Production plans are only accepted by the planners, if they meet certain criteria. These are e.g. that campaigns have to obey a minimal size of 300 tons or that they have to be built of batches with a size of 100 tons each.[14]
- *Production in a chemical plant II*
  Here, a continuous process in a chemical plant that operates 365 days per year and 24 hours per day is to be planned for. The process is interrupted only for maintenance purposes a few days each year. In this process the minimization of changeovers is of paramount importance, because the plant produces off-grade products for a few days after each changeover between two products. Therefore, a minimum length is imposed on each production run. On the other hand,

---

11 Cf. Gopalakrishnan et al. (1995) p. 1974.
12 Cf. Raflatac (2003).
13 Cf. Porkka (2000) pp. 7, 51-57.
14 Cf. Kallrath and Wilson (1997) pp. 303-325 and Kallrath (1999) p. 335.

storage space is limited and costly to increase. Furthermore, the process requires that the plant is always run at a minimum utilization rate.[15]

- *Campaign planning*
  A software company providing supply chain planning software wanted to enhance the modeling and solution capabilities of its mid-term production planning module. In this case it is crucial not only to solve a special case, but to come up with an universal model/algorithm that fits into the architecture of the software system in place. Characteristics within the scope of this project have been an exact modeling of setup operations within a bucket-oriented time structure, specification of minimal campaign lengths and planning of campaigns consisting of batches with fixed size.

1.4 Outline of Thesis

This thesis is organized as follows. In the second chapter basic models in lot-sizing are introduced. According to their underlying time structure they are classified into big-bucket, small-bucket and hybrid models. After having studied the differences of these models in detail, the third chapter analyzes their representation defects with respect to a continuous time scale. Thus, the effect discretization of time imposes on plans, that can be generated by those basic lot-sizing models, is evaluated. This analysis is based on four cornerstones, which are the representation of setup states, the representation of lot sizes, the representation of setup operations and different assumptions on resource utilization.

The fourth chapter provides a thorough literature review which is divided into two parts. The first part reviews basic models in lot-sizing introduced in the second chapter with special emphasis on the extensions to model time continuity as defined in chapter three. The second part contains a review of model formulations originating from applications in the process industries. These often incorporate some aspects of time continuity. As some of these model formulations are based on a continuous time scale, this second part of the literature review is further divided into model formulations based on a discrete representation of time and those based on a continuous representation of time.

The planning framework and techniques considered capable of solving the later proposed model formulations are presented in chapter five. As solution techniques mathematical programming and decomposition will be introduced.

The sixth chapter contains the heart of this thesis. Here, the modeling and solution approach is presented. Mathematical programming model formulations are provided for the four aforementioned aspects to model time continuity in a time-indexed setting (setup states, lot sizes, setup operations and resource utilization). These extensions are given for two different basic models. Furthermore, they are provided as building blocks and may be freely combined dependent on the actual

15 Cf. Lee and Chen (2002) pp. 16-17.

decision situation. Finally, a decomposition heuristic is proposed to allow also for the solution of problems of bigger size.

Computational results are provided in the seventh chapter. These are again organized based on the four aspects to model time continuity in a time-indexed setting (setup states, lot sizes, setup operations and resource utilization). Solutions are analyzed to give insights into what makes certain decision situations difficult. Moreover, computational performance of the proposed model formulations is assessed by comparing them to other model formulations from literature. The extensibility of the proposed model formulations is shown as well as their independence from solver technology.

Finally, chapter eight summarizes the achievements of this thesis and gives a brief outlook on further research opportunities.

2 Basic Models in Lot-Sizing

2.1 Classification of Lot-Sizing Models

Obtaining cost-efficient production plans balancing the trade-off between setup and inventory holding costs – lot-sizing – has been a fundamental goal of practitioners since the beginning of industrialization. The first published work in this area by Harris titled "How many parts to make at once?" dates back as far as to 1913.[16] Since then, a broad stream of research has been developed, dealing with various types of lot-sizing problems for many different applications.

These can be classified according to different attributes.[17] For ease of presentation these attributes are clustered into three sets according to their main relationship: time, resource and product.

The first set "time" contains all attributes with relations to the time structure of the model and the data used:

- *Planning horizon*: The planning horizon may be finite or infinite. Models with an infinite planning horizon usually assume a constant demand rate like the Harris' economic order quantity (EOQ)-model[18] and will not be considered in the remainder.
- *Time scale*: The time scale may either be continuous or discrete. If a discrete time scale is chosen, time buckets may be either big or small and either uniform or non-uniform. Most standard lot-sizing models assume a uniform time discretization, which means that all time buckets have the same size (see sections 2.2 and 2.3). Nevertheless, sometimes a telescopic time scale is chosen with larger time buckets towards the end of the planning horizon[19] or time buckets may be non-uniform for other reasons[20]. The distinction into small- and big-bucket models concerns the relative length of the time periods with respect to the ex-

---

16 Harris (1913).

17 Other compilations of attributes and classifications of lot-sizing models and literature can be found in e.g., Derstroff (1995) pp. 20-24, Domschke et al. (1997) pp. 69-75, Haase (1994) pp. 3-7, Karimi et al. (2003) pp. 366-367, Kuik et al. (1994) pp. 247-249, Meyr (1999) pp. 46-55, Salomon (1991) pp. 21-22, Stadtler (2001) pp. 39-40 and Wolsey (2002) pp. 1589, 1591, 1595.

18 Cf. Harris (1913).

19 E.g., Timpe and Kallrath (2000) pp. 424-425.

20 E.g., Karimi and McDonald (1997) p. 2702.

pected length of any individual production lot.[21] In models with small time buckets it is usually assumed that in each period only one or at most two products may be produced. Therefore small-bucket models integrate lot-sizing and scheduling by not only determining the lot sizes, but also the sequence of production orders. On the other hand, big-bucket models permit multiple products to be produced each period without making any assertion about the sequence of orders.

- *Temporal development of parameters/data*: Parameters can either vary over time (dynamic) or not (static). Often models are distinguished into dynamic or static lot-sizing models according to the temporal development of demands,[22] but generally all parameters (e.g., production capacity, cost parameters, production coefficients) may vary over time.
- *Availability and knowledge of parameters/data*: With respect to the availability and knowledge of the problem data deterministic and stochastic models have to be distinguished. In deterministic models all parameters and data are assumed to be known prior to planning. Stochastic models on the other side try to incorporate the uncertainty of the future into the planning model. This is usually done by assuming a certain distribution or range of values instead of a distinct value for a certain parameter. Typical parameters which are modeled stochastically are e.g. external demands or quantities of defective items.[23] Only deterministic problems will be discussed in the remainder.
- *Objective function*: Most commonly the objective of a lot-sizing problem is to minimize the sum of several cost components. Nevertheless, sometimes other objective functions are defined.[24] These can be either monetary like the maximization of profits or sales, or non-monetary. Then, the goal is not transformed into monetary units, because it is a rather physical accomplishment (e.g., resource leveling) or a temporal objective (e.g., minimization of maximum lateness or total completion time)[25].
- *Cost components*: As stated above the standard objective function is the minimization of several cost components. These comprise classically inventory holding costs and setup costs.

  Inventory holding costs are typically taken into consideration as a linear cost function of the quantity of products in stock at certain points in time. Economically they mainly consist of the costs of capital tied up in inventory. Other parts included in inventory holding costs are costs associated with warehouse operations, taxes, insurance premiums, obsolescence and shrinkage.[26]

---

21 Cf. Salomon (1991) p. 21. Buckets in small-bucket models can have a considerable length depending on the industry and level of aggregation. E.g., De Matta and Guignard (1994) discuss an example of a small-bucket model with a bucket length of one week.

22 Cf. Domschke et al. (1997) p. 70, Kuik et al. (1994) p. 247 and Salomon (1991) p. 21.

23 Cf. Haase (1994) p. 3.

24 Cf. Domschke et al. (1997) p. 73 and Kallrath (2002b) p. 224.

25 Cf. Potts and van Wassenhove (1992) p. 398.

26 Cf. Derstroff (1995) p. 23 and Haase (1994) p. 5.

Setup costs are costs incurred by the production process. Whenever a lot of any product is produced, resources involved in the production process have to be set up to cope with that specific product, e.g., re-tooling of a machine. These costs are charged to the objective function as setup costs. Setup costs consist of direct costs (e.g., cleaning materials) and opportunity costs. Opportunity costs are charged, if setup times, that are the times associated with setup operations, are not considered in the model explicitly. Then, one has to figure out how much capacity has been lost due to the setup operation in order to determine the value of this lost capacity (e.g., contribution margin of those products that could have been produced during the setup operation). Of course, these opportunity costs are hard to estimate as they depend strongly on the scarcity of available capacity which may vary over time and which often is known only after lot-sizing has been done. Therefore, many authors recommend to include setup times into the model and to charge only direct setup costs in the objective function.[27]

Besides these classical cost components of lot-sizing problems other cost components might be considered in the objective function.

First, there are more types of setup related costs. Reservation costs might be charged, if there is a cost associated with preserving the current setup state, when there is no production. Switch-off costs might replace setup costs, if the costs associated with a specific production lot are related primarily to the end of the production process and not to the start (e.g., the main cost component results from cleaning). Generally, it suffices to include either setup or switch-off costs as long as the costs are assumed constant over time and no net present value calculation is performed in the objective function. Furthermore, Wolsey (2002) distinguishes between start-up costs and setup costs, where start-up costs are the costs associated with the start of a production lot and setup costs are charged in each period of production (including the start-up period).[28]

Second, there are also more inventory related costs. These deal with the case, that there is not enough inventory to meet demand. In this case the demand is either lost (lost sales) or fulfilled in later periods (backlog). Both cases are not desirable and therefore a penalty cost is usually associated with these types of "negative" inventories.

Moreover, production costs are most often assumed constant over time and therefore irrelevant in this decision situation. Nonetheless, it might be economically correct to assume declining production costs.[29]

Finally, overtime costs for using extra capacity might be considered in the objective function.

The second set of attributes concerns mainly the "resources" involved in the production planning problem.

27 Cf. Kuik et al. (1994) pp. 249-250 for more criticism to this approach.

28 Cf. Wolsey (2002) p. 1597.

29 Cf. Domschke et al. (1997) p. 72.

- *Capacities*: Capacities of resources can be assumed finite (capacitated) or infinite (uncapacitated). If assumed finite, they might be extended by overtime at a certain cost. Again, this extension can be finite or infinite. Usually capacities may be used up to a fixed budget in each period, e.g., according to a working calendar. On the other hand, very rarely, resources are assumed to be not renewable or only partially renewable. This means the resource availability in a certain period depends on the use of that specific resource in former periods.[30]
- *Product-resource-assignments*: Product-resource-assignments[31] are either such, that each product is assigned to only a single resource (unique assignment) or not.
- *Number of resources*: The planning problem may comprise of one or more resources. If a specific operation can be performed on more than one resource, these resources are called parallel resources. They can be either identical (with respect to the production coefficients, capacities and sets of possible product-resource-assignments) or not. On the other hand, if a single operation requires two or more resources in parallel (e.g., a machine and a worker), we will talk of a problem with multiple resources.
- *Product/operation structure*: The product/operation structure[32] which shows the flow of materials through the production system may be either cyclic or non-cyclic. The product/operation structure is deemed cyclic, if at least one end-product requires two operations at the same resource.
- *Minimal utilization rates*: Minimal production / utilization rates are sometimes taken into account.[33] They are necessary to avoid production plans in which resources are utilized only to a negligible extent. In that case it might be more economical to turn this resource off and shift production to another resource or period.
- *Production coefficient*: Production coefficients are usually deemed constant. That means a production function of Leontief type[34] or linear technology is assumed as a basis. Changes in intensity as considered in the Gutenberg production function[35] are regularly not taken into account, but sometimes the assump-

---

30 Cf. Kimms (1997) pp. 66-68 for an example with partially renewable resources.

31 In this context of planning it is generally not sufficient to examine products at this level of detail. Instead operations should be focused on here, because an operation uses part of the available resource capacity, while a product is usually treated by several operations on (possibly) different resources (e.g., Tempelmeier (2003) p. 207). Nevertheless, we will keep this distinction in mind, but continue to use the terms "product" and "item" as synonyms as done in most of the lot-sizing literature.

32 Cf. Tempelmeier and Helber (1994) pp. 297-298 and Tempelmeier and Derstroff (1996) p. 739.

33 E.g., Kallrath and Wilson (1997) p. 315, Lee and Chen (2002) pp. 21-22 and Wolsey (2002) p. 1597.

34 Cf. Domschke and Scholl (2000) p. 89.

35 Cf. Domschke and Scholl (2000) pp. 92-95, Thommen (1991) pp. 404-407 and Wöhe (1990) pp. 587-594.

tion is made, that resources may operate in different modes.[36] In continuous time model formulations the production coefficients (or the production speed, which is the term usually used in this context) are sometimes assumed zero [$\Rightarrow$production speed: $+\infty$] (e.g., in the EOQ-model).

- *Setup operations*: As already mentioned above where setup related costs were discussed, setup operations are also strongly connected with resources. A setup operation changes the setup state of a resource.

  Setup operations can either be sequence independent or sequence dependent. Moreover, sometimes minor and major setup operations are distinguished. Minor setup operations are setup changes between products of a specific product family, whereas major setup operations are setup changes between products of different product families.[37] Setup times consume part of the available capacity and because the resource is not productive during the setup operation, setup times are reduced as far as possible.[38] As explained above switch-off times might be used instead of setup times.

  In most lot-sizing models setup times are assumed to be small with respect to the bucket length. Therefore setup times are usually modeled such that they lie completely in one period.[39] On the other hand, this is often not true for small-bucket models. Consequently, setup times may span over several periods.[40] Furthermore, setup operations can be confined to start only at the beginning of a period (e.g. on Monday morning) or they may lie during downtimes (e.g. on weekends) to save productive time.

The last set of attributes "products" is devoted to the output of the production process.

- *Number of products*: The number of products considered is an important attribute. Single-product problems and multi-product problems have to be distinguished.
- *Bill of materials (BOM) structure*: The number of levels in the BOM and the structure of the BOM lead to two more attributes:

  According to the number of levels, single-level and multi-level problems are distinguished. In single-level problems there is no relationship between products in form of a BOM.

  Regarding the structure of the BOM several cases are usually distinguished. In an assembly structure each (non-end) product has exactly one successor in the BOM. In a divergent structure each (non-purchase) product has exactly one predecessor in the BOM. A serial structure illustrates the special case of inheriting an assembly structure as well as a divergent (arborescent) structure. A gen-

---

36 Cf. Kallrath and Wilson (1997) pp. 310-311.

37 Cf. Potts and van Wassenhove (1992) p. 397.

38 Cf. Trigeiro et al. (1989) p. 353.

39 Cf. Grünert (1998) pp. 47-48.

40 Cf. Drexl and Haase (1995) pp. 81-82, Haase (1994) pp. 31-35 and Helber (1994) pp. 34-38.

eral structure can contain both elements as sub-structures (assembly and divergent structure). Fig. 2-1 shows examples of the different product structures.

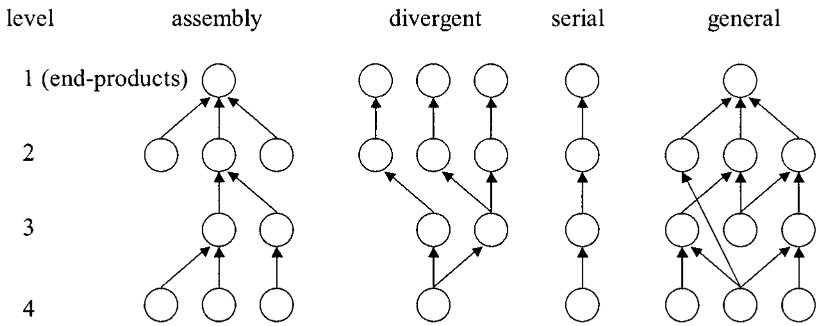


Fig. 2-1: Typical structures of BOMs.

- *Supply process*: The supply process concerns the issue, at which time a product becomes available to further stages/levels after being processed at one stage. The first option is, that each single item of a lot will be available immediately to the next stage of processing (e.g., on a flow line). Other options are, that products become available only if the full lot has been finished, a certain batch has been finished or after a defined lead time.
- *Lead times*: Lead times in a lot-sizing model can be either endogenous or exogenous. Endogenous lead times are the result of the model and therefore determined by the solution procedure. On the other hand, exogenous lead times, which are given by the circumstances of the production process (e.g., transportation time between successive resources or simply drying of paint), result in minimal lead times that have to be respected.[41] Even if no exogenous lead time applies, it may be necessary to request a minimal lead time of one period. Otherwise the resulting production plan may not be feasible.[42] Fig. 2-2 shows an example: Products 3 and 4 are predecessors of products 1 and 2. Although products 3 and 4 are procured in the desired quantity to produce 1 and 2 in period t, this schedule is technically not feasible, as product 4, which is needed as an input for 1 is produced too late. Whether product 3 (as an input to product 1) can be procured in time depends on the assumptions made regarding the supply process. Anyway, if a minimal lead time of one period was requested beforehand, products 3 and 4 would have been produced in period t–1 at the latest.

---

41 Cf. Kuik et al. (1994) p. 248.

42 Cf. Grünert (1998) pp. 48-49 and Haase (1994) pp. 14-18.

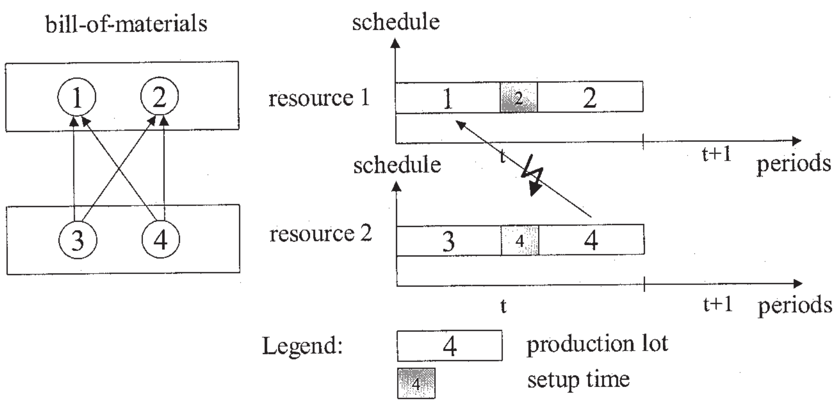


Fig. 2-2: Example for minimal required lead time to ensure feasibility.

- *Inventory restrictions*: Sometimes restrictions on inventory levels of products are imposed. These can be upper bounds to model a warehouse capacity or lower bounds to ensure that a certain safety stock is always held.[43] Moreover, shelf-life restrictions might be included to prevent obsolescence of products. Shelf-life restrictions are common in the food industries.[44]
- *Service policy*: Service policies concern demand fulfillment. In most lot-sizing models an a priori known demand has to be fulfilled. As explained earlier, there are also models, in which demand not met is either lost (lost sales) or can be fulfilled in subsequent periods (backlog).[45] Another possibility is that demand is assumed adjustable within certain bounds. Then the model has to decide which demands are met and to which extent.[46]
- *Additional lot-sizing rules*: Additional lot-sizing rules may apply. In the process industries the production quantity of a certain lot is often restricted. These restrictions can be lower bounds or upper bounds on the lot size or that the lot size has to be in multiples of a predefined batch size.[47] These kind of restrictions will be introduced in more detail in section 3.3. Other rules are minimum on and off times.[48] These imply that if a certain product is produced it has to be produced for a certain (minimal) time (minimum on-time). On the other hand, if production has switched to the next product, the first product must not be scheduled again for a certain (minimal) time (minimum off-time).

---

[43] Cf. Belvaux and Wolsey (2001) pp. 996, 1001, 1004.
[44] E.g., Brown et al. (2001) p. 9.
[45] Cf. Haase (1994) p. 6.
[46] Cf. Wolsey (2002) p. 1591.
[47] Cf. Kallrath (1999) p. 334, Kallrath (2002b) pp. 224-225 and Suerie (2004) pp. 3-4.
[48] Cf. Belvaux and Wolsey (2001) p. 1000.

In the remainder of this thesis the distinction between big-bucket and small-bucket models will play an important role. Furthermore, a third class of models (hybrid models) will be introduced, which encompasses models with elements of big-bucket as well as small-bucket models. The basic models of these three classes will be introduced briefly in the following three subsections.

Models will be judged on their ability to model a continuous time scale. Therefore the following characteristics will be measured for the introduced models. The first one will be the *preservation of setup states*, whereas the second one will be the *lot size* (or campaign size, see section 1.2) a certain model permits. Lot size is hereby defined as the quantity of a distinct product that is produced after a setup operation for this product has been performed. Generally speaking, models are judged on the impact of time discretization on resulting production plans.

2.2 Big-Bucket Models: Capacitated Lot-Sizing Problem

The Capacitated Lot-Sizing Problem (CLSP) is the most basic big-bucket model studied in the context of multi-item capacitated lot-sizing.[49] Its fundamental assumptions are:

- Several products j are produced on one shared resource.
- The resource has a capacity limit.
- The planning horizon is finite and divided into T periods.
- All products face a deterministic dynamic demand.
- If a product is produced in a certain period, the resource has to be set up for this product in this period.
- Setups consume resource capacity and incur a setup cost.
- The aim is to minimize the sum of holding costs and setup costs.

Mathematically, the CLSP can be stated as follows.

Model CLSP:

$$\text{Min} \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} h_{jt}\cdot I_{jt} + \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} sc_j \cdot Y_{jt} \qquad (2\text{-}1)$$

$$I_{jt-1} + X_{jt} = d_{jt} + I_{jt} \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T} \qquad (2\text{-}2)$$

$$\sum_{j\in\mathcal{J}} a_j \cdot X_{jt} + \sum_{j\in\mathcal{J}} st_j \cdot Y_{jt} \le c_t \qquad \forall\ t\in\mathcal{T} \qquad (2\text{-}3)$$

$$X_{jt} \le b_{jt}\cdot Y_{jt} \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T} \qquad (2\text{-}4)$$

$$X_{jt} \ge 0,\quad I_{jt}\ge 0,\quad I_{j0}=0 \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T} \qquad (2\text{-}5)$$

---

49 Cf. among others Dixon and Silver (1981) p. 24, Maes and van Wassenhove (1988) pp. 991-992, Richter (1975) pp. 385-386 and Trigeiro et al. (1989) pp. 354-355.

$$Y_{jt} \in \{0;1\} \qquad \forall j \in \mathcal{J},\ t \in \mathcal{T} \qquad (2\text{-}6)$$

Indices and index sets:

j	Products or items, $j \in \mathcal{J}$
t	Periods, $t \in \mathcal{T}$
$\mathcal{J}$	Set of products
$\mathcal{T}$	Set of periods

Data:

a_j	Capacity consumption to produce one unit of item j (=production coefficient)
b_{jt}	Large number, not limiting feasible production quantities of product j in period t
c_t	Available capacity in period t
d_{jt}	Primary, gross demand for item j in period t (with d_{jT} including final inventory, if given for the planning horizon T)
h_{jt}	Holding cost for one unit of product j in period t
sc_j	Setup cost for product j
st_j	Setup time for product j

Variables:

I_{jt}	Inventory of item j at the end of period t
X_{jt}	Production quantity of item j in period t (lot size)
Y_{jt}	Setup variable (=1, if a setup operation for item j is performed in period t, =0 otherwise)

The objective function (2-1) aims at minimizing the sum of holding costs and setup costs. Constraints (2-2) are inventory balance constraints and ensure that all demands are met in time. Available capacity may be used up by production of items (X_{jt}) and setup operations (Y_{jt}) due to constraints (2-3). Constraints (2-4) couple production variables X_{jt} with setup operations Y_{jt}, where b_{jt} is defined such, that it is not limiting feasible lot sizes of product j. Finally, (2-5) and (2-6) impose non-negativity conditions and binary conditions, respectively.

The CLSP is a big-bucket model. So far, it inherits no aspects of time continuity, neither within the individual buckets nor between them: Within each individual bucket no sequence of lots is given. Across time buckets no information is passed from one period to the next, except for the inventories. Therefore, the buckets are totally decoupled regarding the production process. Consequently, setup states are not preserved across periods. This means, a setup operation for a certain product j is necessary at the beginning of period t, even if this product was produced last in period t–1, because this information is not available in the model.

Furthermore, due to the model formulation the lot sizes in the resulting production plan are limited by available capacity per period.[50]

[50] To be exact, they are limited for product j in period t by $(c_t - st_j)/a_j$.

2.3 Small-Bucket Models

In contrast to big-bucket models, small-bucket models consider lot-sizing and sequencing decisions simultaneously.[51] Therefore, at most one setup operation is allowed in each period. Different model formulations with various degrees of freedom with respect to the decision variables of the model are known in this area.

2.3.1 Discrete Lot-Sizing and Scheduling Problem

The Discrete Lot-Sizing and Scheduling Problem (DLSP) is the model formulation with the least degree of freedom of the models presented here. Its fundamental assumptions are:[52]

- Several products j are produced on one shared resource.
- The resource has a capacity limit.
- The planning horizon is finite and divided into T periods.
- All products face a deterministic dynamic demand.
- Only one product can be produced in each period.
- If a product is produced in a period, it will be produced at full capacity during this period (all-or-nothing assumption).
- Setup operations incur a setup cost. Thereby, a setup operation reflects the change of the setup state of a resource.
- The aim is to minimize the sum of holding costs and setup costs.

Mathematically, the DLSP can be stated as follows.

Model DLSP:

$$\text{Min} \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} h_{jt}\cdot I_{jt} + \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} sc_j \cdot Y_{jt} \qquad (2\text{-}7)$$

$$I_{jt-1} + \frac{c_t}{a_j} Z_{jt} = d_{jt} + I_{jt} \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T} \qquad (2\text{-}8)$$

$$\sum_{j\in\mathcal{J}} Z_{jt} \le 1 \qquad \forall\ t\in\mathcal{T} \qquad (2\text{-}9)$$

$$Y_{jt} \ge Z_{jt} - Z_{jt-1} \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T} \qquad (2\text{-}10)$$

$$Y_{jt} \ge 0, \quad I_{jt} \ge 0, \quad I_{j0} = 0 \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T} \qquad (2\text{-}11)$$

$$Z_{jt} \in \{0;1\}, \quad Z_{j0} = 0 \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T} \qquad (2\text{-}12)$$

---

51 See section 2.1.

52 Cf. Fleischmann (1990) pp. 337-338 and Magnanti and Vachani (1990) p. 458.

Variables: [53]

Z_{jt} Setup state variable (=1, if item j is setup at the end of period t, =0 otherwise)

The DLSP aims at minimizing the sum of holding and setup costs (2-7). As production has to be at full capacity or not at all in each period, the model formulation does not rely on production variables X_{jt}, which are replaced by $(c_t/a_j)\cdot Z_{jt}$ in the inventory balance constraints (2-8). Constraints (2-9) and (2-10) couple variables that indicate a setup operation Y_{jt} and variables that indicate the setup state at the end of each period Z_{jt}. At most one setup state can exist at the end of a specific period (2-9). If the setup state changes in two consecutive periods, a setup operation Y_{jt} must have taken place (2-10). Setup costs in the objective function are costs related to these setup operations. (2-11) and (2-12) impose non-negativity and binary conditions on the decision variables.

Due to the fundamental assumptions of the model, the DLSP inherits first aspects of time continuity. Setup states are preserved across periods.[54] This means production of a specific product j can continue in period t without performing a new setup operation, if j was also produced in t–1. Therefore, lots can extend over several periods and are therefore only bounded by the maximal production capacity within the planning interval. On the contrary, lot sizes are still restricted due to the all-or-nothing assumption.

2.3.2 Continuous Setup Lot-Sizing Problem

The Continuous Setup Lot-Sizing Problem (CSLP)[55] abandons the all-or-nothing assumption of the DLSP. Apart from that, its fundamental assumptions are the same as stated for the DLSP.

Formally, the CSLP can be stated as follows.

Model CSLP:

$$\text{Min} \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} h_{jt}\cdot I_{jt} + \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} sc_j \cdot Y_{jt} \qquad (2\text{-}13)$$

$$I_{jt-1} + X_{jt} = d_{jt} + I_{jt} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \quad (2\text{-}14)$$

$$X_{jt} \le \frac{c_t}{a_j} Z_{jt} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \quad (2\text{-}15)$$

53 In the remainder indices, index sets, data or variables will be explained within the text only at their first occurrence. A complete list of symbols is given in the List of Symbols.

54 In the model formulation presented here, setup states are still lost in idle periods. But there are other model formulations for the DLSP, in which the setup state is preserved even during idle periods (e.g., Magnanti and Vachani (1990) pp. 458-459).

55 Cf. Karmarkar and Schrage (1985) pp. 328-329 and Salomon (1991) pp. 33-36.

$$\sum_{j \in \mathcal{J}} Z_{jt} \leq 1 \qquad \forall\, t \in \mathcal{T} \quad (2\text{-}16)$$

$$Y_{jt} \geq Z_{jt} - Z_{jt-1} \qquad \forall j \in \mathcal{J},\, t \in \mathcal{T} \quad (2\text{-}17)$$

$$X_{jt} \geq 0, \quad Y_{jt} \geq 0, \quad I_{jt} \geq 0, \quad I_{j0} = 0 \qquad \forall j \in \mathcal{J},\, t \in \mathcal{T} \quad (2\text{-}18)$$

$$Z_{jt} \in \{0;1\}, \quad Z_{j0} = 0 \qquad \forall j \in \mathcal{J},\, t \in \mathcal{T} \quad (2\text{-}19)$$

The objective function aims at minimizing the sum of holding and setup costs (2-13). (2-14) are inventory balance constraints. In constraints (2-15) production variables X_{jt} are linked with setup state variables Z_{jt}, whereas (2-16) and (2-17) couple the setup operation variables Y_{jt} and setup state variables Z_{jt} as has been done in the DLSP ((2-9) and (2-10)). Finally, (2-18) and (2-19) impose non-negativity and binary conditions on the decision variables. Setup times are easily integrated into the model formulation by adding CLSP-like capacity constraints (2-3).[56]

With respect to time continuity, the CSLP resembles the DLSP. In contrast to the DLSP the setup state is even preserved during idle periods, leading to a more natural representation of the shop floor. Furthermore, lot sizes can take arbitrary values now, because the all-or-nothing assumption has been dropped.

Solutions to the CSLP will always have a better (or equal) objective function value than solutions to the DLSP of the same test instance because of the excess production implied by the all-or-nothing assumption of the DLSP. On the other hand, the lower degree of freedom allows to develop more efficient solution procedures for the DLSP than for the CSLP.[57] So both models have their eligibility in respective situations.

2.3.3 Proportional Lot-Sizing and Scheduling Problem

The Proportional Lot-Sizing and Scheduling Problem (PLSP)[58] is the most versatile of the small-bucket models. It shares the same basic assumption with the DLSP and CSLP, apart from the all-or-nothing assumption and the restriction to produce only one product per period. The latter is revoked by allowing one setup operation per period. Thereby, two products may be produced each period: One before and one after the setup operation has taken place. Moreover, the sequence of lots is still determined by the model, but the capacity of a period, which is lost in the CSLP when there is no production at full capacity, can be utilized to set up and produce another product.

Mathematically, the PLSP can be stated as follows.

---

56 Cf. Hauth (1998) pp. 53-56.

57 Cf. Meyr (1999) pp. 64-65.

58 Cf. Haase (1994) pp. 26-27, Drexl and Haase (1995) pp. 74-75, Kimms and Drexl (1998) pp. 1196-1198 and Belvaux and Wolsey (2001) p. 999.

Model PLSP:

$$\text{Min} \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} h_{jt}\cdot I_{jt} + \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} sc_j \cdot Y_{jt} \tag{2-20}$$

$$I_{jt-1} + X_{jt} = d_{jt} + I_{jt} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \tag{2-21}$$

$$\sum_{j\in\mathcal{J}} a_j \cdot X_{jt} + \sum_{j\in\mathcal{J}} st_j \cdot Y_{jt} \le c_t \qquad \forall\, t\in\mathcal{T} \tag{2-22}$$

$$X_{jt} \le \frac{c_t}{a_j}\cdot (Z_{jt} + Z_{jt-1}) \qquad \forall\, j\in\mathcal{J}, t\in\mathcal{T} \tag{2-23}$$

$$\sum_{j\in\mathcal{J}} Z_{jt} \le 1 \qquad \forall\, t\in\mathcal{T} \tag{2-24}$$

$$Y_{jt} \ge Z_{jt} - Z_{jt-1} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \tag{2-25}$$

$$X_{jt} \ge 0, \quad Y_{jt} \ge 0, \quad I_{jt} \ge 0, \quad I_{j0} = 0 \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \tag{2-26}$$

$$Z_{jt} \in \{0;1\}, \quad Z_{j0} = 0 \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \tag{2-27}$$

Again, the objective is to minimize the sum of holding and setup costs (2-20). (2-21) are the regular inventory balance constraints and (2-22) are capacity constraints, stating that production and setup operations never exceed available capacity. Constraints (2-23) link production variables X_{jt} with setup state variables Z_{jt}. Production of j in t is allowed, if the product j is set up either at the beginning of t (=at the end of $t-1$, $Z_{jt-1}=1$) or at the end of t ($Z_{jt}=1$). (2-24) take care that there is at most one setup state at the end of each period and (2-25) force the setup operation variable Y_{jt} to "1", if the setup state is changed within a period. (2-25) together with (2-20) guarantee that at most one setup operation is performed in each period and therefore the sequence of products is determined by this model formulation. Finally, (2-26) and (2-27) impose non-negativity and binary conditions, respectively.

The solution space of the PLSP surpasses the solution space of the CSLP, because it is able to utilize the capacity lost in the CSLP, when there is no full production in certain periods.[59] Regarding time continuity, setup states are preserved during idle periods and lot sizes can take arbitrary values in the PLSP. The only restrictions still imposed by time discretization are, that setup operations must lie completely within a period due to constraints (2-22)[60], and – of course – the limit of one setup operation per period.

---

[59] Cf. Haase (1994) p. 30 for a comparison of the set of feasible solutions of the small-bucket models DLSP, CSLP and PLSP and Kimms and Drexl (1998) p. 1198.

[60] Drexl and Haase (1995) pp. 81-82, Haase (1994) pp. 31-35 and Helber (1994) pp. 34-38 propose model formulations of the PLSP in which even setup times can extend over several periods.

2.4 Hybrid Models

Apart from the big-bucket and small-bucket models introduced in sections 2.2 and 2.3, there are three other models that do not fit into this classification. These are models with a big-bucket structure, i.e. which allow to produce multiple products per period, but on the other hand, give information (at least partially) on the sequence of products, i.e. allow to preserve setup states across periods.

2.4.1 Capacitated Lot-Sizing Problem with Linked Lot Sizes

The first model of this kind is the Capacitated Lot-Sizing Problem with Linked Lot Sizes (CLSPL).[61] It represents an extension to the CLSP by allowing to carry over a setup state across one or more periods. Therefore, it shares the fundamental assumptions with the CLSP except that it adds the link option.

Formally, it can be stated as follows (with objective function (2-1) and constraints (2-2), (2-3) and (2-5) taken from the CLSP).

Model CLSPL:

$$\sum_{j \in \mathcal{J}} W_{jt} \leq 1 \qquad \forall\, t \in \mathcal{T} \setminus \{1\} \qquad (2\text{-}28)$$

$$W_{jt} \leq Y_{jt-1} + W_{jt-1} \qquad \forall j \in \mathcal{J},\, t \in \mathcal{T} \setminus \{1\} \qquad (2\text{-}29)$$

$$W_{jt+1} + W_{jt} \leq 1 + V_t \qquad \forall\, j \in \mathcal{J},\, t \in \mathcal{T} \setminus \{T\} \qquad (2\text{-}30)$$

$$Y_{jt} + V_t \leq 1 \qquad \forall\, j \in \mathcal{J},\, t \in \mathcal{T} \qquad (2\text{-}31)$$

$$X_{jt} \leq b_{jt} \cdot (Y_{jt} + W_{jt}) \qquad \forall j \in \mathcal{J},\, t \in \mathcal{T} \qquad (2\text{-}32)$$

$$V_t \geq 0, \quad V_1 = 0, \quad V_T = 0 \qquad \forall\, t \in \mathcal{T} \setminus \{T\} \qquad (2\text{-}33)$$

$$Y_{jt} \in \{0;1\}, \quad W_{jt} \in \{0;1\}, \quad W_{j1} = 0 \qquad \forall j \in \mathcal{J},\, t \in \mathcal{T} \qquad (2\text{-}34)$$

Variables:

V_t Single-product indicator, which indicates that (a) only one product is produced in period t, and (b) the setup state is carried over from period $t-1$ and carried into the next period $t+1$ (V_t=1); otherwise (V_t=0)

W_{jt} Link variables, which indicate that product j was scheduled last in period $t-1$ and therefore production can continue in period t without performing a new setup operation in period t (W_{jt}=1); otherwise (W_{jt}=0)

---

[61] Cf. Dillenberger et al. (1993) pp. 106-113, Haase (1994) pp. 18-21, Gopalakrishnan et al. (1995) pp. 1975-1981 and Suerie and Stadtler (2003) pp. 1041-1044.

Constraints (2-28) ensure that at most one setup state can be preserved from one period to the next. Variables W_{jt} therefore correspond to variables Z_{jt-1} of the small-bucket models introduced in section 2.3 by representing the setup state at the intersection of two adjacent periods. (2-29) guarantee that a setup state can be carried into period t only if either a setup operation has been performed in period $t-1$ ($Y_{jt-1}=1$) or the setup state has already been carried over into period $t-1$ ($W_{jt-1}=1$). The same setup state is allowed on two consecutive period boundaries only, if the single-product indicator variable V_t is equal to "1" (2-30). This is only possible if no setup operation is performed in the enclosed period t (2-31).[62] Constraints (2-32) allow production of a certain item j in period t, if either a setup operation is performed in period t ($Y_{jt}=1$) or the setup state is carried into period t from period $t-1$ ($W_{jt}=1$). (2-33) and (2-34) impose non-negativity and binary conditions on the decision variables. Due to constraints (2-30) and (2-31) variables V_t will only take binary values in feasible solutions of model CLSPL without being defined binary explicitly.

The set of feasible solutions to the CLSPL includes both, the set of feasible solutions to the CLSP and the set of feasible solutions to the PLSP. The CLSP is derived from the CLSPL by forbidding the linking option, i.e. all variables W_{jt} are set to zero, whereas the PLSP is derived from the CLSPL by limiting the number of setup operations in each period to one for all periods. This constraint would have to be added. Therefore, the CLSPL is the most powerful model introduced so far.

With respect to time continuity the CLSPL allows arbitrary lot sizes and a preservation of setup states across (idle) periods. The last restriction, that prevents it from allowing to model all solutions that might be possible on a continuous timescale are constraints (2-3), which force setup operations to lie completely within a period.

2.4.2 Capacitated Lot-Sizing Problem with Sequence Dependent Setup Costs

The Capacitated Lot-Sizing Problem with Sequence Dependent Setup Costs (CLSD) is an extension to the CLSPL, as it allows to model sequence dependent setup costs (and times).[63] Other than that, it shares the same fundamental assumption as the CLSP and CLSPL. To yield the correct sequence dependent setup costs and times it is assumed that the so-called triangle condition is always fulfilled.

The triangle condition states that it is more expensive (consumes more time) to change the setup state from product j to product i via product k than to change directly from j to i.

---

62 The special case, that the same setup state is carried over on consecutive period boundaries *and* a setup operation is performed in the enclosed period is also possible within this CLSPL model formulation. Cf. Suerie and Stadtler (2003) p. 1053 for an example.

63 Cf. Grünert (1998) pp. 49-56, Haase (1996) pp. 53-55 and Meyr (1999) pp. 68-71.

$$sc_{ji}^{sd} \leq sc_{jk}^{sd} + sc_{ki}^{sd}$$
$$st_{ji}^{sd} \leq st_{jk}^{sd} + st_{ki}^{sd} \qquad \forall\, i,j,k \in \mathcal{J} \quad (2\text{-}35)$$

Model CLSD can be stated as follows.

Model CLSD:

$$\text{Min} \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} h_{jt}\cdot I_{jt} + \sum_{i\in\mathcal{J}}\sum_{\substack{j\in\mathcal{J}\\ j\neq i}}\sum_{t\in\mathcal{T}} sc_{ij}^{sd}\cdot Y_{ijt}^{sd} \qquad (2\text{-}36)$$

$$I_{jt-1} + X_{jt} = d_{jt} + I_{jt} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \quad (2\text{-}37)$$

$$\sum_{j\in\mathcal{J}} a_j\cdot X_{jt} + \sum_{i\in\mathcal{J}}\sum_{\substack{j\in\mathcal{J}\\ j\neq i}} st_{ij}^{sd}\cdot Y_{ijt}^{sd} \leq c_t \qquad \forall\, t\in\mathcal{T} \quad (2\text{-}38)$$

$$X_{jt} \leq b_{jt}\cdot\Big(\sum_{\substack{i\in\mathcal{J}\\ i\neq j}} Y_{ijt}^{sd} + Z_{jt-1}\Big) \qquad \forall\, j\in\mathcal{J}, t\in\mathcal{T} \quad (2\text{-}39)$$

$$\sum_{j\in\mathcal{J}} Z_{jt} = 1 \qquad \forall\, t\in\mathcal{T} \quad (2\text{-}40)$$

$$\sum_{\substack{i\in\mathcal{J}\\ i\neq k}} Y_{ikt}^{sd} + Z_{kt-1} = \sum_{\substack{j\in\mathcal{J}\\ j\neq k}} Y_{kjt}^{sd} + Z_{kt} \qquad \forall\, k\in\mathcal{J}, t\in\mathcal{T} \quad (2\text{-}41)$$

$$F_{jt} \geq F_{it} + 1 - J\cdot(1 - Y_{ijt}^{sd}) \qquad \forall\, i\in\mathcal{J}, j\in\mathcal{J}\setminus\{i\}, t\in\mathcal{T} \quad (2\text{-}42)$$

$$X_{jt}\geq 0,\quad F_{jt}\geq 0,\quad I_{jt}\geq 0,\quad I_{j0}=0 \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \quad (2\text{-}43)$$

$$Y_{ijt}^{sd}\in\{0;1\},\quad Z_{jt}\in\{0;1\},\quad Z_{j0}=0 \qquad \forall\, i\in\mathcal{J}, j\in\mathcal{J}, t\in\mathcal{T} \quad (2\text{-}44)$$

Indices and index sets:

i, *k* Products or items, $i\in\mathcal{J}$, $k\in\mathcal{J}$

Data:

sc_{ij}^{sd} Sequence dependent setup cost, if a setup operation from product *i* to product *j* is performed

st_{ij}^{sd} Sequence dependent setup time, if a setup operation from product *i* to product *j* is performed

Variables:

F_{jt} Position variable (takes only integer values), the larger F_{jt} the later product *j* is scheduled in period *t*

Y_{ijt}^{sd} Sequence dependent setup variable (=1, if a setup operation from item *i* to item *j* is performed in period *t*, =0 otherwise)

The objective function of the CLSD is to minimize holding costs and sequence dependent setup costs (2-36). Constraints (2-37) are the standard inventory balance constraints and (2-38) are capacity constraints, which take sequence dependent setup times into account. (2-39) allow production of item j in period t, if a setup operation originating from any other product i to product j is performed ($\sum_{i \in J} Y_{ijt}^{sd} = 1$) or the setup state for item j is carried into period t from period t-1 (Z_{jt-1}=1). At the end of each period, a certain setup state must exist (2-40). Constraints (2-41) describe the flow of setups. The left-hand side (LHS) is equal to "1", if a setup operation for product k is performed in period t ($\sum_{i \in J} Y_{ikt}^{sd} = 1$) or the setup state for product k is carried into period t (Z_{kt-1}=1). Then, the setup state for k has to be carried into period t+1, and thus Z_{kt}=1, or a setup operation from product k to any other product j is performed in period t ($\sum_{j \in J} Y_{kjt}^{sd} = 1$). The same reasoning holds, if either side equals zero. Finally, (2-42) eliminates all possible subtours.[64] (2-43) and (2-44) impose non-negativity and binary conditions.

To obtain the sequence of products in a big-bucket period, sequence dependency has to be introduced into the formulation. With this extension it is possible to obtain the sequence of products within each period and thus lot-sizing and sequencing is done simultaneously like in small-bucket models. Note, that with non-sequence dependent data, the CLSD solves the same problem as the CLSPL, but gives the sequence within periods at the (computational) cost of a lot more binary variables.[65]

Again, the restriction, that setup operations lie completely in one period, is the only type of constraints that prevents the CLSD from allowing to model all production plans that might be possible on a continuous time-scale (2-38).

2.4.3 General Lot-Sizing and Scheduling Problem

The last model considered here is the General Lot-Sizing and Scheduling Problem (GLSP).[66] The GLSP is the most general model (and therefore justifies its name) in the sense that it includes all models considered so far as special cases. Its fundamental assumptions are:

- Several products j are produced on one resource.
- The planning horizon is finite and divided into T (big-bucket) periods.
- Each big-bucket period t is divided into a set of small-bucket periods s.
- The resource has a capacity limit in each of the big-bucket periods t.
- All products face a deterministic dynamic demand based on the big-bucket periods.
- In each small-bucket period s at most one product j is produced.
- Setup states can be carried over across (small-bucket and big-bucket) periods.
- Setup operations consume resource capacity and incur a setup cost.

64 Cf. Haase (1996) p. 54 and Miller et al. (1960) p. 327.

65 Of course, it does not make any sense to use the model CLSD for data that is not sequence dependent, because then the sequence within periods can be arbitrary. Anyhow, this comparison illustrates the relationship of models CLSPL and CLSD.

66 Cf. Fleischmann and Meyr (1997) pp. 12-13 and Meyr (1999) pp. 78-82.

- The aim is to minimize the sum of holding costs and (sequence dependent) setup costs.
- The triangle condition (on setup costs and setup times, (2-35)) does not have to hold and therefore the number of setup operations per big-bucket period is not restricted by the number of products.

Mathematically, the GLSP can be stated as follows.

Model GLSP:

$$\text{Min} \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} h_{jt}\cdot I_{jt} + \sum_{i\in\mathcal{J}}\sum_{\substack{j\in\mathcal{J}\\ j\neq i}}\sum_{t\in\mathcal{T}} sc_{ij}^{sd}\cdot Y_{ijt}^{sd} \tag{2-45}$$

$$I_{jt-1} + \sum_{s\in\mathcal{S}_t} X_{js} = d_{jt} + I_{jt} \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T} \tag{2-46}$$

$$\sum_{s\in\mathcal{S}_t}\sum_{j\in\mathcal{J}} a_j\cdot X_{js} + \sum_{s\in\mathcal{S}_t}\sum_{i\in\mathcal{J}}\sum_{\substack{j\in\mathcal{J}\\ j\neq i}} st_{ij}^{sd}\cdot Y_{ijs}^{sd} \le c_t \qquad \forall\ t\in\mathcal{T} \tag{2-47}$$

$$X_{js} \le \frac{c_t}{a_j} Z_{js} \qquad \forall\ j\in\mathcal{J},\ t\in\mathcal{T},\ s\in\mathcal{S}_t \tag{2-48}$$

$$\sum_{j\in\mathcal{J}} Z_{js} = 1 \qquad \forall\ t\in\mathcal{T},\ s\in\mathcal{S}_t \tag{2-49}$$

$$Y_{ijs}^{sd} \ge Z_{is-1} + Z_{js} - 1 \qquad \forall\ i\in\mathcal{J}, j\in\mathcal{J}\backslash\{i\},\ t\in\mathcal{T},\ s\in\mathcal{S}_t \tag{2-50}$$

$$X_{js} \ge minlot_j\cdot(Z_{js} - Z_{js-1}) \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T},\ s\in\mathcal{S}_t \tag{2-51}$$

$$X_{jt}\ge 0,\quad I_{jt}\ge 0,\quad I_{j0}=0 \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T} \tag{2-52}$$

$$Y_{ijs}^{sd}\ge 0,\quad Z_{js}\in\{0;1\},\quad Z_{j0}=0 \qquad \forall\ i\in\mathcal{J}, j\in\mathcal{J},\ t\in\mathcal{T},\ s\in\mathcal{S}_t \tag{2-53}$$

Indices and index sets:

s	(Small-bucket) periods, $s\in\mathcal{S}_t$
$\mathcal{S}_t$	Set of (small-bucket) periods that form (big-bucket) period t

Data:

$minlot_j$	Minimal lot size for product j

The objective function aims at minimizing holding costs and sequence dependent setup costs (2-45). Constraints (2-46) are inventory balance constraints. The sum in (2-46) reflects that two different time scales are used in this model. The first time scale denoted by the index t corresponds to a big-bucket period (or macro-period as it is called in Fleischmann and Meyr (1997)[67]). Inventory variables I_{jt} and demands d_{jt} are only updated according to this time scale. Within each big-

[67] Cf. Fleischmann and Meyr (1997) pp. 11-21.

bucket period t a set of small-bucket periods (micro-periods) S_t is defined. Capacity constraints (2-47) are based on big-bucket periods t, because the capacity assigned to each small-bucket period is a decision of the model. Therefore, capacity constraints state, that production and setup operations of all small-bucket periods s, which belong to big-bucket period t ($s \in S_t$), will not exceed available capacity in period t. (2-48) couple production quantities X_{js} and binary setup state variables Z_{js}. (2-49) take care that the setup state at the end of each small-bucket period is well-defined. Constraints (2-50) couple setup operation variables Y_{ijs}^{sd} with setup state variables Z_{js}. As it is assumed explicitly, that the triangle condition does not have to hold, constraints (2-51) have to be introduced into the model. They enforce a minimal production on each product j that has been set up to avoid that this setup is chosen to circumvent a direct setup change without actually producing ($i \rightarrow j \rightarrow k$, instead of $i \rightarrow k$).[68] (2-52) and (2-53) are non-negativity conditions and binary conditions. Although not defined as binary variables explicitly, variables Y_{ijs}^{sd} will only take binary values in feasible solutions due to (2-45) and (2-50).

Compared to the CLSD, the GLSP will have the same optimal solution, if the triangle condition (2-35) holds. Otherwise the GLSP is more flexible, as it allows multiple setup operations for each product j in each big-bucket period t.

With respect to time continuity, again, setup times have to lie completely within one (big-bucket) period (2-47). Furthermore, this is also true for minimal lot sizes. Therefore, the lifting of the triangle condition leads to a new restriction regarding the modeling of solutions on a continuous time scale.

2.5 Relationship Between Models

From the description of the seven models which have been introduced in subsections 2.2–2.4, it is obvious, that these models have more in common than separates them. The following discussion of the models will focus on one model from each group: one big-bucket model (CLSP), one small-bucket model (PLSP) and one hybrid model (CLSPL). The PLSP is preferred over the DLSP and CSLP, because it contains the largest amount of freedom, as it is neither restricted by an all-or-nothing assumption like the DLSP nor by the limit of one product per period like the CSLP. As one aim of this thesis will be to derive production plans, that match production plans derived on a continuous time scale best, the PLSP seems to be the one most suited to provide an appropriate basic model. The CLSPL is chosen among the hybrid models, because the CLSD and GLSP assume sequence dependency among setups, which will not be a topic in this thesis. Therefore, these two models picture more detail than required here.

---

68 The triangle condition does not hold, if the intermediate product j has a cleaning effect on the resource. However, to utilize this cleaning effect, the intermediate product j has to be produced for a certain minimal amount ($minlot_j$). E.g., Fleischmann and Meyr (1997) p. 13 and Meyr (2002) p. 280.

Table 2-1 classifies the CLSP, PLSP and CLSPL according to the attributes presented in section 2.1. The planning horizon is finite, data and parameters are assumed dynamic and deterministic and the objective is the minimization of setup and holding costs in all three models. With respect to the time scale, the period length gives rise to identify big buckets at the CLSP and CLSPL and small buckets at the PLSP. Nevertheless, the CLSPL has not been introduced as a big-bucket model, but rather as a hybrid model, as it allows the preservation of setup states across periods (analogue to the PLSP), which is constituent to small-bucket models. All three models are capacitated, assume constant production coefficients and sequence independent setup times. No differences regarding attributes concerning products can be observed between the three models. All of them are formulated as single-level problems here, assume only endogenous lead times, no inventory restrictions, a demand fulfillment service policy and no additional lot-sizing rules.

Table 2-1: Classification of CLSP, PLSP and CLSPL.

	CLSP	PLSP	CLSPL
Time			
Planning horizon	←	finite	→
Time scale	discrete, big buckets	discrete, small buckets	discrete, big buckets
Temporal development of parameters/data	←	dynamic	→
Availability/knowledge of parameters/data	←	deterministic	→
Objective function	←	minimization of costs	→
Cost components	←	setup costs and holding costs	→
Resources			
Capacities	←	capacitated	→
Product-resource assignment	←	(unique)	→
Number of resources	←	1	→
Product/operation structure		–	
Minimum utilization rates	←	none	→
Production coefficient	←	constant	→
Setup operations	←	sequence independent setup times	→
Setup state preservation across periods	no	yes	yes
Products			
Number of Products	←	multiple	→
Bill-of-materials structure	←	single-level	→
Supply process		–	
Lead times	←	endogenous	→
Inventory restrictions	←	none	→
Service policy	←	demand fulfillment	→
Additional lot-sizing rules	←	none	→

A short example, which illustrates the modeling capabilities of the CLSP, PLSP and CLSPL, respectively, will guide the discussion of the advantages and disadvantages of these models. In this example the aim is to plan for the production of three different products, which share a common resource with limited capacity over four periods. Two of the products face a rather steady demand, whereas the third product is requested only sporadically. Table 2-2 gives the data of the problem.

Table 2-2: Example data.

Product	Demand d_{jt}				Production coefficient	Setup time	Capacity	Holding cost (per period)	Setup cost
j	t=1	t=2	t=3	t=4	a_j	st_j	c_t	h_j	sc_j
j=1	20	25	30	35	1	10		0.5	10
j=2	0	25	30	35	1	10	80	0.5	10
j=3	0	0	0	10	1	10		0.5	10

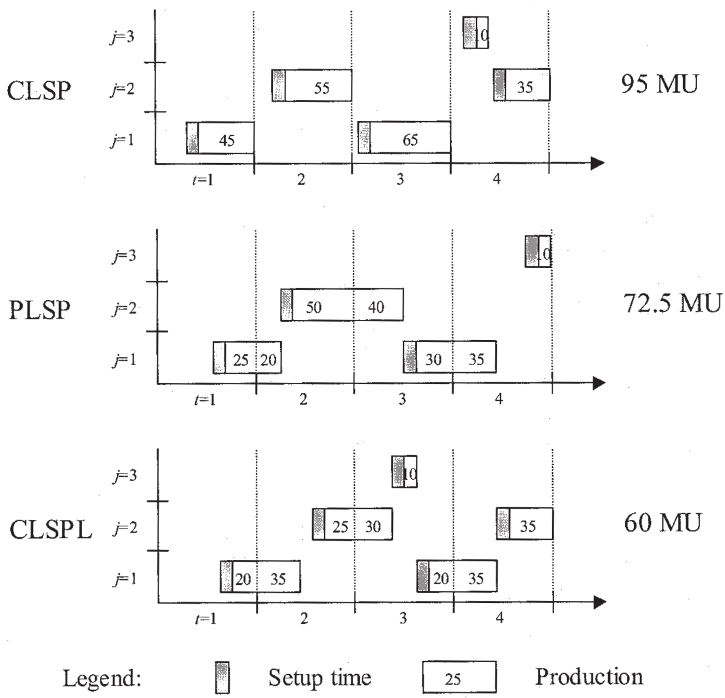


Fig. 2-3: Optimal solutions of CLSP, PLSP and CLSPL.

Fig. 2-3 shows the optimal solutions to this problem for all three model formulations. The objective function values are 95, 72.5 and 60 monetary units [MU] for the CLSP, PLSP and CLSPL, respectively. The solutions are shown as gantt charts.[69]

[69] In general, solutions to the CLSP and CLSPL are not well-defined regarding the sequence within periods, because this information is not available in the model formula-

The CLSP possesses the largest objective function value. As setup state preservation across periods is not possible here, large lot sizes within periods are built leading to high holding costs. Anyhow, the solution may be modified by a planner if a continuous time scale was intended. By right-shifting of the first three lots and therefore allowing an ex-post preservation of setup states, total costs of the plan may be reduced by 30 [MU] to 65 [MU] due to the reduction of inventory holding costs.[70]

The PLSP allows the preservation of setup states and therefore saves much of the holding costs compared to the CLSP by splitting production of a certain lot into adjacent periods. Regarding lot sizes and the sequence of lots, the production plan created by the PLSP resembles the CLSP plan. Only the lot size for the production of product 2 differs between the two plans.

The CLSPL shows the lowest objective function value. This observation does not hold for this example only, but also in general. Compared to the CLSP, the CLSPL can always generate the same solution or a better solution, if it is possible to link at least two lot sizes of adjacent periods, which saves ceteris paribus a setup operation and thus setup costs. Compared to the PLSP, the CLSPL can again generate the same solutions, but also better ones, if it is advantageous to perform two setup operations in a certain period.

In this example the CLSP suffers from the non-preservation of setup states, whereas the PLSP suffers from the restriction, that only one setup change is allowed in each period. The CLSPL on the other hand benefits from the fact, that exactly these two restrictions are not existent in its model formulation and therefore provides the best plan of the three. In general, the CLSPL is able to combine the advantages of both models.

The following deficiencies of the CLSP as a typical big-bucket model are put forward:[71]

- Sequencing aspects are disregarded.
- Aggregation of small-bucket periods may lead to infeasibilities.
- In multi-level problems long processing times result due to the long period length.

On the other hand, the following deficiencies of the PLSP are often mentioned:[72]

- No buffers against uncertainty can be kept, because the resulting plan is too detailed.
- Freedom to change elements of the plan is very limited.

---

tion. In this example, this issue arises only in period four of the CLSP solution, where an arbitrary sequence is chosen.

70 15 units of the third lot are moved into period 4 (saving 7.5 [MU]), the second lot is moved 20 units into period 3 (saving 10 [MU]) and the first lot is moved 25 units into period 2 (saving 12.5 [MU]).

71 Cf. Haase (1994) pp. 11-18.

72 Cf. Derstroff (1995) pp. 32-36 and Helber (1994) pp. 38-40.

- Rolling schedules will result in plan nervousness due to frequent changes of data.

The advantages of the CLSP and PLSP are the counter arguments of the other model. On the other hand, the CLSPL is universal, inheriting both other models as special cases. Therefore, its character can be shifted either towards a small- or big-bucket model depending on the period length, which is chosen in a certain situation. Thus, the CLSPL is the basic model of choice from a modeling perspective. However, the CLSPL is also the most demanding model from a computational perspective.

The modeling of time continuity, which is the main topic focused in this thesis, is only possible, if adjacent periods can be linked somehow. Therefore we will focus on the CLSPL and PLSP as basic models in the remainder of this thesis. Nevertheless, we keep in mind, that the PLSP due to its restriction to allow at most one setup operation per period, imposes a hard constraint on potential solutions, whereas the CLSPL is in general preferable due to its universality.

3 Extensions to the Basic Models: Time Continuity

After having reviewed the basic models of lot-sizing in chapter 2, the extensions considered in this thesis will be introduced in detail here. The focus will be on the representation of arbitrary (continuous) plans on a discrete time scale. Therefore, the time structure of the models in general is analyzed first. Three building blocks corresponding to representation defects due to time discretization are taken into consideration thereafter: the carry-over of setup states, lots spanning over several periods and setup operations spanning over several periods. In conjunction with these extensions constraints regarding the utilization of resources might become an issue. Consequently, this topic is also discussed.

3.1 Time Structure

One of the attributes that has been used to classify lot-sizing models in section 2.1 is the time scale. This topic is analyzed in more detail here.

Different factors influencing the time scale are often distinguished. Timpe and Kallrath (2000) distinguish between a commercial time scale and a production time scale due to the different requirements of marketing and production.[73] Furthermore, Meyr (1999) and Pressmar (1980) differentiate into an exogenous (outside) and an endogenous (inside) time structure.[74] Generally speaking, the choice of an appropriate time scale for production planning depends a lot on the inside dynamics of the production system, that is planned for, as well as the outside dynamics of the world the production system is embedded into.

The exogenous (outside) time structure is usually out of the sphere of influence. Fig. 3-1 shows the impact of demand variation on different time scales as an example. External demands cannot be influenced by the company and are therefore likely to determine constraints on the time structure, which has to be used. If demand is stable, this poses no restrictions on the appropriate time scale. In this case, models working on a continuous time scale with an infinite planning horizon are often chosen.[75] The counterpart of this scenario is a discrete time structure which implies discrete demand variation. There, the demands are bundled such, that for each time bucket only an aggregated demand for the specific period needs to be

---

[73] Cf. Timpe and Kallrath (2000) pp. 424-425.

[74] Cf. Meyr (1999) pp. 49-51 and Pressmar (1980) pp. 455-456.

[75] See also section 2.1.

considered. In between these two extreme cases lies a free time structure, which either rests on continuous or discrete demand variation. Demand variation may either be discrete, e.g., if certain demands are the result of fixed customer orders at certain due dates, or continuous, e.g., if it is the result of a continuous demand function in a multi-level production process.[76]

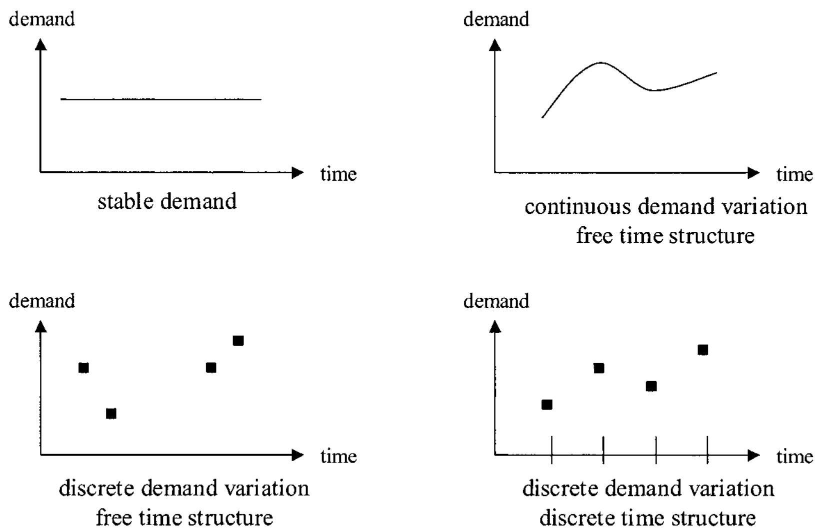


Fig. 3-1: Impact of demand variation on different time scales.[77]

The endogenous time structure, which is the time structure used for production planning, is chosen by the planner. It has to respect the implications of the exogenous time structure to which it is usually coupled as well as the internal dynamics of the production system.[78] Consequently, the endogenous time structure, i.e. the time scale of the production planning model, is often a finer resolution of the exogenous time structure.[79]

In the remainder continuous changes, which are mainly in the scope of short-term scheduling, will be disregarded. Instead, as a mid-term view is chosen, some aggregation is deemed possible, such that in reality continuous processes may be represented as discrete processes without too much loss of detail. However, the discrete time scale has to be chosen carefully.[80] Moreover, a discrete time scale

---

76 Cf. Meyr (1999) pp. 49-51.

77 Cf. Meyr (1999) p. 50.

78 Cf. Meyr (1999) pp. 50-51 and Pressmar (1980) pp. 455-457.

79 E.g., Fleischmann (1994) p. 396 argues that the external demands are not based on the periods of his small-bucket DLSP model, but on larger macroperiods, which each contain a (different) number of microperiods. This is also true for the GLSP introduced in section 2.4.3.

80 Cf. Kuik et al. (1994) p. 247.

has also other advantages in terms of interpretability of results and planning of other side constraints (e.g., personnel availability, if personnel is not modeled as a resource explicitly). The discrete time buckets might be days, weeks or shifts, so the workload can be read off right from the planning result.

Anyhow, as we will see in sections 3.2 to 3.4, the discretization of time imposes some unwanted restrictions on the solution space and even prohibits certain solutions, which would have been possible, if a continuous time scale was used. Fig. 3-2 shows an example of the representation defect of the CLSP compared to a representation of the solution on a continuous time scale (two setups for C). How to overcome these representation defects due to time discretization will be the main topic of this thesis.

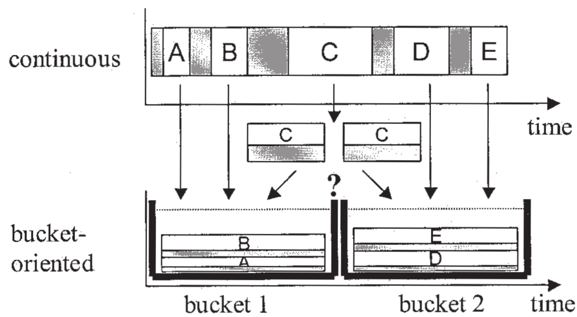


Fig. 3-2: Representation defect of a bucket-oriented model formulation (here: CLSP).

Last but not least, regarding the discretization of time, several rules have to be obeyed depending on the type of model chosen for production planning. If the model, e.g. an CLSP[81], is based on the assumption, that only one lot of each product may be produced each period, this imposes also a restriction on the period length, which has to be chosen such that it is not economical to produce a product in two or more lots in a certain period.[82]

The next three sections will emphasize three extensions of standard lot-sizing models that incorporate – with increasing detail – aspects of time continuity within a time-bucket oriented setting.

3.2 Setup States

When thinking about time continuity in time-bucket oriented lot-sizing model formulations, the first issue that comes into mind, is the preservation of setup states across periods or setup carry-over, as it is sometimes referred to.[83] Preserva-

81 See section 2.2.

82 Cf. Bogaschewsky (1988) p. 168.

83 E.g., Gopalakrishnan et al. (1995), Sox and Gao (1999) and Porkka et al. (2003).

tion of setup states is already included in most of the basic models introduced in chapter 2. It is common knowledge for small-bucket models, because it is one of their constituent attributes. Moreover, the hybrid models proposed in section 2.4 also contain setup carry-over. Nevertheless, as preservation of setup states is an important first step in representing time continuity in time-bucket oriented lot-sizing models, the modeling of setup state preservation will be motivated here.

Fig. 3-3 shows in a didactic example, what happens, if preservation of setup states is not explicitly modeled. The first gantt chart depicts the case, in which setup times and setup carry-over are not modeled. Accordingly, the result is that setup costs are accounted for twice and available capacity is overestimated. The second gantt chart on the other hand depicts the case, where setup times are modeled, but setup carry-over is still out of the scope of the model formulation. Again, the result is not satisfactory. Still setup costs (and now also setup times) are accounted for twice and worse, available capacity is now underestimated, which results in an infeasible production plan here, because the gray product cannot be fully produced by period *t* due to the shortcomings of the model formulation. Only the third gantt chart captures the characteristics of the real world and provides a realistic capacity allocation.

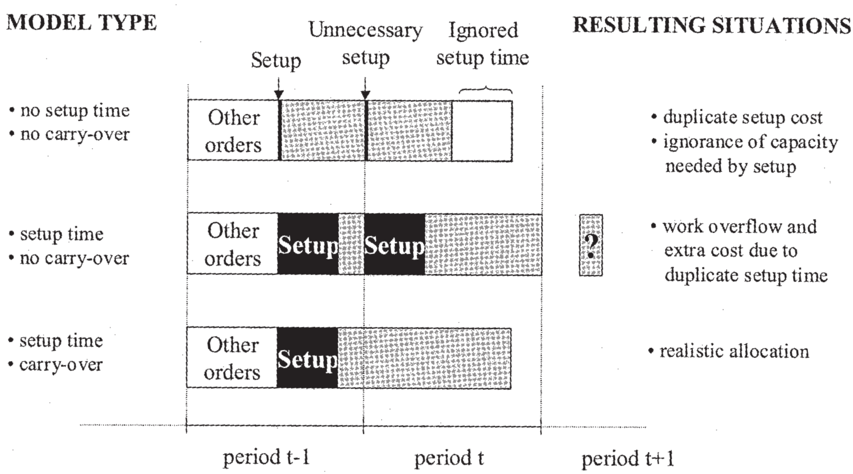


Fig. 3-3: Motivating example for preservation of setup states.[84]

Infeasible production plans due to the underestimation of available capacity are not only found in small didactic examples, but also in real world applications.[85] Especially, if capacity utilization is high, small savings in allocated setup times

84 Porkka et al. (2003), p. 1135.

85 E.g., Gopalakrishnan et al. (1995), pp. 1981-1985. Moreover, Smith-Daniels and Ritzman (1988) pp. 664-665 stress the importance of setup carry-over for process industries.

can make the difference between feasibility and infeasibility of a solution of a production planning problem.

Based on the results of Porkka et al. (2003), it can be concluded that a considerable amount of capacity is freed, if setup carry-over is included in the model formulation.[86] They conducted an experimental study to evaluate the effectiveness of setup carry-over. Table 3-1 shows results derived from their study. In their test set setup times for all products were equal for each test instance and varied between 2 % and 15 % of the constant capacity per period. The capacity has been calculated such, that it matches on average 75 %, 100 % respectively 110 % of the capacity that would have been needed, if a lot-for-lot production was assumed (lot-for-lot utilization rate). Table 3-1 shows the difference in slack capacity, if a CLSPL is used for modeling instead of a CLSP. As expected, freed capacity increases, if setup times are relatively longer. This effect is even bolstered up, if capacity is tighter restricted.

Surprisingly, not as much capacity is freed for additional production as one would have expected. If all setup times are equal to 15 % of available capacity and in each period one setup carry-over is utilized,[87] one would expect that 15 % of capacity are freed. This is clearly not the case.

Table 3-1: Freed capacity as a percentage of total capacity depending on the relative setup time and lot-for-lot utilization rate. (Based on the results of Porkka et al. (2003), p. 1144, table 5.)

Lot-for-lot utilization rate	Relative setup time			
	2 %	5 %	10 %	15 %
75 %	0.6 %	1.6 %	2.3 %	3.5 %
100 %	1.0 %	1.8 %	3.3 %	4.0 %
110 %	1.0 %	2.8 %	3.8 %	5.6 %

Why? If less capacity is freed and the same amount of products is produced, then the CLSPL must allocate relatively more available capacity to setup operations. Here, roughly 2/3 of theoretically freed capacity are re-allocated to new setup operations. This leads to the presumption, that plans created by a CLSPL model formulation are fundamentally different from those created by a CLSP model formulation. This is a strong argument for incorporating preservation of setup states into the lot-sizing model formulation.

Consequently, several authors have made comparisons of (modified) CLSP and CLSPL solutions.[88]

Haase (1998) concludes from his computational tests, that the inclusion of setup carry-over into a special purpose heuristic leads to significantly better solutions than the modification of CLSP solutions by a simple heuristic. He further

---

86 Cf. Porkka et al. (2003) p. 1144.

87 Cf. Porkka et al. (2003) p. 1143. Porkka (2000) p. 28 reports only one exception.

88 Cf. Gopalakrishnan et al. (2001) p. 861, Haase (1998) pp. 140-143, Porkka et al. (2003) pp. 1141-1146 and Sox and Gao (1999) pp. 176-178. Modified means, that some authors altered CLSP solutions by a simple heuristic to contain setup carry-over.

observes that this effect is more important the less products compete against each other for scarce capacity. Anyhow, as his test set comprises instances with J=8, 20 and 50 products, this result is quite obvious.[89]

Sox and Gao (1999) compare solutions of CLSP, a modified CLSP and CLSPL. They find out that modified CLSP solutions, although these compare very favorably to CLSP solutions, are still more expensive than CLSPL solutions. Moreover, they have developed a measure to account for the difference in plans created by different solution procedures. This measure calculates the fraction of total demand that is produced in different periods in the two plans, that are to be compared. In their test set, which comprises of only five test instances, this fraction amounts to 26.44%.[90]

Gopalakrishnan et al. (2001) compare the objective function values of a test set solved with a special purpose CLSP heuristic with the solutions of their tabu-search CLSPL heuristic. From their extensive computational tests they conclude that the incorporation of setup carry-over leads to substantial savings. Again this effect is more concise the less products compete for scarce capacity.[91]

Finally, Porkka et al. (2003) have conducted the most comprehensive study on the difference of CLSP, modified CLSP and CLSPL solutions. Regarding objective function values, they confirm the results of previous studies. Modified CLSP solutions, although better than (optimal) plain CLSP solutions, are by far more expensive than optimal CLSPL solutions. In their test set, CLSPL solutions show less setup costs as well as less inventory holding costs than respective CLSP solutions.[92] Furthermore, they have analyzed the solutions based on the difference measure proposed by Sox and Gao (1999)[93]. According to them, 47 % of production is allocated to different periods in CLSP and CLSPL solutions, respectively.

All these results strongly support that the preservation of setup states across periods is an important task as an enhancement of lot-sizing models in general, if necessitated by the production system.

3.3 Lot Sizes

After having motivated why it is important to couple the production in adjacent periods via setup states, now the lot size itself will be focused on. In the process industries, there often arises the problem, that a lower and/or upper bound is imposed on a continuous production run or that production has to be in multiples of a predefined batch size.[94] This problem is usually referred to as campaign planning, with a campaign defined as the production amount of a specific product type of

89 Cf. Haase (1998) p. 142.
90 Cf. Sox and Gao (1999) pp. 176-178.
91 Cf. Gopalakrishnan et al. (2001) p. 861.
92 Cf. Porkka et al. (2003) pp. 1141-1146.
93 See definition above or Sox and Gao (1999) p. 177.
94 Cf. Kallrath (2002b) pp. 224-225 and Plapp (2003) p. 20.

one continuous production run, which can and generally will span over several planning periods.

This task is illustrated in the following example (Table 3-2 and Fig. 3-4),[95] which is a slightly modified version of the example used in section 2.5 to demonstrate the difference of the presented model types. Here, the example comprises a single machine, two products and four periods. The PLSP, which has been introduced in section 2.3.3, will serve as a basic model. From the data given in Table 3-2 follows a simple lot-for-lot production as the optimal solution with an objective function value of 40 monetary units [MU], when the standard PLSP model is used for optimization (first gantt chart, PLSP). Therefore, the total costs only consist of setup costs (4·10) and no inventory holding costs.

Now, campaign production comes into play. As campaign restrictions may be fulfilled in periods after the planning horizon of four periods, campaign restrictions are not enforced on the last campaign within the planning interval in this small example.

Table 3-2: Data for illustrative example.[96]

Product j	Demand d_{jt} t=1	t=2	t=3	t=4	Production coefficient a_j	Setup time st_j	Available capacity c_t	Holding cost (per period) h_j	Setup cost sc_j
j=1	0	25	30	35	1	10	80	1	10
j=2	20	25	30	35	1	10		1	10

First, minimal campaign lengths will be looked at. Minimal campaign length means, that there exists a minimum production amount, that has to be produced, whenever a lot of a certain product is started.[97] Typically, a minimal campaign length is process dependent, e.g., a critical mass is required to initiate a chemical reaction. When minimum production amounts per campaign of, e.g., 50 units are brought into the example, the PLSP plan will no longer be feasible, because the first campaign of product j=2 lasting from period t=1 to t=2 yields only 45 units. Fig. 3-4 (second gantt chart, MIN) shows the optimal solution to this slightly modified problem. Five units of production are shifted from the second campaign of product j=2 to the first campaign at an additional cost of holding five units of inventory for one period (period t=2 to t=3).

Second, instead of a lower bound on the campaign length an upper bound on the production amount per campaign (maximal campaign length) is introduced. The rationale behind this is that, e.g., a cleaning operation may be required every time that a certain amount has been produced.[98] In the example this upper bound is assumed to be 60 units. Again the initial PLSP solution is not feasible as the second campaign for product j=2 lasting from period t=3 to t=4 yields 65 units. The

95 Cf. Suerie (2004) pp. 2-4.

96 Cf. Suerie (2004) p. 2.

97 E.g., Braun (2002) p. 14, Kallrath (1999) pp. 334-335, Lee and Chen (2002) pp. 21-22 and Porkka (2000) p. 62.

98 E.g., Kallrath (1999) p. 334.

optimal solution to this slightly modified problem is shown in the third gantt chart (Fig. 3-4, MAX). Here, the shifting of five units stems from the violation of the upper bound restriction of the second campaign of product j=2.

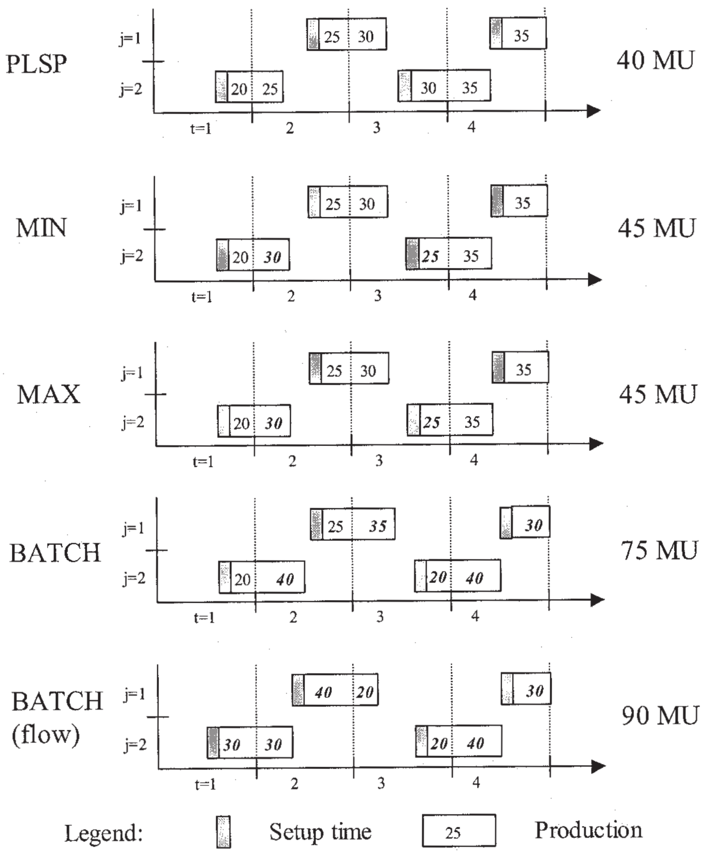


Fig. 3-4: Solution to illustrative example.[99]

In a third scenario, production must be in multiples of a predefined batch size (here: 20 units).[100] Batch size restrictions often arise in the process industries, where for example the batch size is determined by a reactor load. Again, the initial PLSP solution needs several modifications to comply with this additional constraint. The fourth gantt chart (Fig. 3-4, BATCH) shows the optimal solution with an objective function value of 75 [MU]. Here, it is not sufficient only to shift production quantities. Excess production of ten units of product j=2 is required as the

---

99 Cf. Suerie (2004) p. 3.

100 E.g., Braun (2002) p. 14, Kallrath (1999) pp. 334-335 and Porkka (2000) p. 62.

total demand within the planning interval (110 units) for this product is not a multiple of the batch size.

The fourth and last scenario stems from the fact that in campaign production different assumptions regarding the flow of materials have to be distinguished. So far it has been assumed that any production is available immediately, that is continuous flow. In batch production often not only the total production quantity has to be in multiples of a batch size, but also the flow of materials is in batches (e.g., tanks). Therefore, materials only become available after the full batch size is produced. This phenomenon is also called batch availability.[101] If this restriction applies, the solution shown in the fourth gantt chart will no longer be feasible with respect to product j=1 and period t=2. Although 25 units are produced here, which matches the demand, five units will only become available when the second batch of product j=1 is finished in period t=3. The optimal solution to this problem is shown in the last gantt chart (Fig. 3-4, BATCH (flow)).

Looking at this simple example it seems obvious how to modify optimal PLSP plans to take into account campaign restrictions. Nevertheless, in general time discretization forces a strict separation of production amounts belonging to different periods, albeit they may form one lot together, when watched on a continuous time scale. The information that the first lot of product j=2 (Fig. 3-4, PLSP) comprises of 45 units is not available from the model, but rather the information that 20 units are produced in period t=1 and 25 units in t=2. Consequently, it is not straightforward to include a constraint into the model formulation, which restricts the production amount of a campaign.

If the campaign restrictions were based on the time-indexed variables of the standard (PLSP) model formulation instead, this would distort production plans and unnecessarily limit the solution space. For example, minimal campaign length would have to lie completely within one period. The optimal solution to the above example would have shown the same lot sizes as in the second gantt chart (Fig. 3-4, MIN), but the first three campaigns would have been left-shifted to lie completely within one period. This would result in an objective function value of 140 [MU] (45+30+30+35).

Generally speaking, if plans that are feasible on a continuous time scale must be matched by a model formulation working on a discrete time scale, the model formulation has to take into account that lot sizes may comprise of production in several adjacent periods.

3.4 Setup Operations

When setup states and lot sizes are allowed to span over several periods, the last attribute not modeled with respect to time continuity is that setup operations might be spread over two or more periods. Up to now this feature was totally out of

101 Cf. Potts and Kovalyov (2000) pp. 228, 247.

scope of big-bucket models.[102] In conjunction with small-bucket models this feature has been tackled as an extension to the DLSP, but there setup times have to be in multiples of a period's length.[103] Moreover, several authors presented a mathematical model formulation of the PLSP with the extension to allow setup times of arbitrary length.[104] Unfortunately, none of these very versatile PLSP extensions is mathematically correct. All of them fail even on small test instances,[105] which should be motivation enough to develop a new – mathematically correct – model formulation.

Anyhow, there are more arguments justifying a closer look at this topic. First, albeit much effort has been taken to reduce setup times, these still can consume a substantial portion of available capacity.[106]

Second, even if setup times are operationally reduced, it has been noticed that their relative share of capacity roughly stays the same, because as soon as the reduction has been achieved, setups are done more frequently.[107] This is also supported by the observation in section 3.2 (Table 3-1), that two thirds of the expected increase in capacity go into new setup operations.

Finally, we will consider a thought experiment. Starting point is a production plan based on a continuous time scale. Setup times account for ten percent of capacity in this experiment. If the time scale is divided into two periods, the chance, that exactly the same optimal solution can be obtained from a bucket-oriented model is 90 %, because the probability that a setup operation is performed at the bucket boundary is ten percent. Usually the time scale is not divided into two, but rather a lot more periods. If there were ten periods to be planned for instead, the probability, that the same optimal solution can be obtained with a standard bucket-oriented formulation decreases to 39 %. If setup times had been 15 % of capacity,

---

[102] E.g., Grünert (1998) pp. 47-48.

[103] Cf. Cattrysse et al. (1993) pp. 477-478, Salomon (1991) pp. 86-87 and Salomon et al. (1991) pp. 805-806.

[104] Cf. Drexl and Haase (1995) pp. 81-82, Haase (1994) pp. 31-35 and Helber (1994) pp. 36-38.

[105] The table depicts such a test instance which none of the model formulations by Drexl and Haase (1995) pp. 81-82, Haase (1994) pp. 31-35 and Helber (1994) pp. 36-38 is able to solve to optimality correctly. These model formulations fail, because (a) setup times might be attributed to two setup operations at a time and/or (b) binary values might allow production in certain constellations without any setup time attributed to this setup.

Product j	Demand d_{jt} t=1	t=2	t=3	t=4	Production coefficient a_j	Setup time st_j	Available capacity C_t	Holding cost (per period) h_j	Setup cost sc_j
j=1	0	20	0	30	1	100	80	2	50
j=2	0	0	20	0	1	60		2	50

[106] E.g., Hindi et al. (2003) p. 490, Leschke (1995) p. 12 and Trovinger and Bohn (2003) p. 2.

[107] E.g., Porkka et al. (2003) p. 1133.

it would have decreased to 23 %.[108] Although this does not give any indication, how good or bad the resulting plan of a bucket-oriented model formulation would be, it seems worthwhile to examine this case.

Moreover, if setup operations can be modeled to span over two or more periods, together with the extensions proposed in sections 3.2 and 3.3, any plan that can be represented on a continuous time scale may also be generated in a bucket-oriented setting.

3.5 Resource Utilization

Although not related directly to modeling on a continuous time scale, constraints regarding resource utilization may become an issue in the models considered in this thesis. When setup carry-over is introduced into a model formulation, this feature is almost always used in an optimal solution. Unfortunately, this will often lead to solutions in which machines run continuously (to save setups), but well below capacity. However, this is not a desired outcome and will not likely find the approval of the decision maker involved.

Therefore, constraints forcing a minimal (and/or maximal) resource utilization (or production rate) come into play. Several scenarios are possible:

- *Minimal utilization*: Minimal utilization rates are common in the process industries. The argumentation closely follows the one above regarding minimal campaign lengths (section 3.3). Mostly, minimal utilization rates are induced by the processes. Consequently, several authors have proposed model formulations which contain this feature.[109]
- *Full utilization*: A special case is to force full utilization of capacity. This has also been a constituent characteristic of one of the small-bucket models introduced in section 2.3, namely the DLSP. Moreover, an extension to the CLSPL, which forces full capacity utilization whenever production is run, has been presented.[110]
- *Constant utilization*: Constant capacity utilization is required e.g. in paper production.[111] It is a relaxation of the full utilization case, but a further restriction with respect to the minimal utilization case, because here production output is not allowed to vary for a certain lot.

Moreover, combinations of the above scenarios are possible (e.g., minimal and constant utilization). Apart from that, constant or minimal utilization does not only

---

108 The capabilities of the bucket-oriented model formulation are rated too low by this rough estimate, if setup times can be (left-)shifted due to available capacity, but are intended to give an indication.

109 E.g., Kallrath (1999) p. 332, Kallrath and Wilson (1997) p. 315 and Lee and Chen (2002) pp. 21-22.

110 Cf. Porkka et al. (2003) p. 1138.

111 Cf. Porkka (2000) p. 11 and Porkka et al. (2003) p. 1138.

make sense for resources, but although for the workforce, where a constant workload is even more important.

A further distinction arises, if machines cannot be switched off. In this case the e.g. minimal production rates must be met throughout the whole planning interval, whereas otherwise the constraints on resource utilization only have to hold whenever actual production takes place.

Summing up the discussion, the modeling of resource utilization is an important issue when necessary, but closely dependent on the process which is modeled.

4 Literature Review

The aim of this literature review is twofold. The first aim is to give a brief overview on lot-sizing models focusing on the extensions presented in chapter 3. This is done in the first part of this chapter (section 4.1). A more general review of lot-sizing models is omitted, as several of these are broadly available from literature.[112] Only hybrid models are reviewed in some more detail, because these are in the focus of this thesis. In the second part (section 4.2) production planning models originating from specific process industries problems are examined. Although often many problem specific characteristics are incorporated in these models, they are of intense interest here, because some modeling ideas might be borrowed from these models and – even more important – they constitute potential areas of application for the models proposed in this thesis.

4.1 Basic Models

The representation of time in lot-sizing models is crucial for the integration of the proposed extensions which have been presented in chapter 3. As has been emphasized in the classification of lot-sizing models (section 2.1), most of them operate on a discrete time scale with either uniform or non-uniform time buckets. These may be further distinguished by the nature of the time buckets in either small or big-bucket models. Bucket-oriented models will be reviewed in the following sections.

A model formulation related to a small-bucket model (CSLP) which contains some noteworthy characteristics with respect to the time domain is proposed by Pressmar (1980).[113] The modeling of time is discussed in detail and resulting is a model with time buckets of variable length. These time buckets of variable length are coupled to some static time domain by a second time scale, which is given by exogenous events (e.g., due dates for orders) of the planning environment. Therefore, this model formulation in a sense combines a continuous time scale with a

---

[112] See Domschke et al. (1997) pp. 69-165 for a comprehensive overview on all kinds of lot-sizing problems and solution algorithms, Kuik et al. (1994) pp. 243-263 for a broad review and classification of lot-sizing models, Drexl and Kimms (1997) pp. 221-235 or Staggemeier and Clark (2001) pp. 938-947 for recent surveys focusing on small-bucket and hybrid models and Karimi et al. (2003) pp. 369-375 for a recent review on solution approaches for the CLSP as well as the literature cited in these references.

[113] Cf. Pressmar (1980) pp. 458–466.

discrete time scale. Unfortunately, to make the problem tractable for MIP optimization, the objective function needs to be linearized, implicitly assuming a uniform time discretization again.

Continuous time scales are assumed by lot-sizing models which belong to the Economic Lot Scheduling Problem (ELSP) family. The ELSP is based on the assumptions that several products are produced on one resource in a cyclical pattern. Production and demand rates for all products are constant and demands have to be met. Setup costs and times have to be considered and the tradeoff between setup costs and inventory holding costs is to be optimized.[114]

The assumption of a constant demand rate is exploited by the fact that a cyclical production pattern can be generated as a solution. Cyclical production plans have the advantage of minor plan nervousness and easier integration to other stages of the order fulfillment process (e.g., purchasing and distribution) as well as other areas of a company (e.g., finance).[115] On the other hand, this defining principle poses a hard constraint, and consequently this type of model is not applicable, if dynamic demands are present. Therefore, ELSP-like models, for which efficient algorithms have been developed,[116] will not be considered in the remainder.

4.1.1 Big-Bucket Models

The CLSP is the most basic lot-sizing problem.[117] Its fundamental assumptions and model formulation have been presented in section 2.2. Even the single-product version of the CLSP is known to be NP-hard,[118] and the same is true for finding feasible solutions if setup times are present.[119] Numerous surveys are available for this problem, with the most recent survey by Karimi et al. (2003) focusing on different solution approaches (exact methods, common-sense / specialized heuristics and mathematical programming-based heuristics).[120]

The CLSP (without setup times) is studied by Constantino (1998). In his paper minimal production quantities are analyzed. In contrast to minimal campaign quantities the minimal production quantities that are studied there have to be respected in each period of production. The paper contains a polyhedral study and derives valid inequalities and a separation algorithm to deal with this kind of problem.[121]

---

114 Cf. Elmaghraby (1978) pp. 587-588.

115 Cf. Carstensen (2002) pp. 2.

116 E.g., Dobson (1987) pp. 764-771, Zipkin (1991) pp. 56-63, Carstensen (2002) pp. 53-79 and Wagner and Davis (2002) pp. 133-146.

117 The uncapacitated single-product version is known as the Wagner-Whitin problem for which polynomial time solution algorithms exist. Cf. Wagner and Whitin (1959) p. 93 and Wagelmans et al. (1992) pp. S147-S152.

118 Cf. Florian et al. (1980) pp. 670-676 and Bitran and Yanasse (1982) pp. 1180-1183.

119 Cf. Maes et al. (1991) pp. 135-136.

120 Cf. Kuik et al. (1994) pp. 258-259 and Karimi et al. (2003) pp. 369-375.

121 Cf. Constantino (1998) pp. 101-118.

Furthermore, minimal and maximal production quantities per period are also present in the model formulation tackled by Belvaux and Wolsey (2000) by means of a specialized branch-and-cut code.[122]

4.1.2 Small-Bucket Models

Small-bucket models have been in the scope of lot-sizing research for more than 20 years. Drexl and Kimms (1997) have provided a comprehensive survey focusing on this topic. Complexity considerations show that even the basic DLSP is NP-hard.[123]

Blocher et al. (1999) propose an algorithm for a DLSP-like problem. In contrast to the basic DLSP, their objective is to minimize setup costs. Setup times are also considered, but are restricted to be in multiples of a period's length.[124] Furthermore, the basic DLSP with period overlapping setup times, which have to be multiples of a period's length, is considered by Cattrysse et al. (1993) as well as by Brüggemann and Jahnke (1994).[125]

The DLSP in various mutations is solved by Jordan and Drexl (1998). They relate the DLSP to a scheduling problem, which they term batch sequencing problem. With their solution algorithm for the batch sequencing problem they solve the DLSP with sequence independent setup times and costs, sequence dependent setup costs as well as with sequence dependent setup costs and times. Setup times can be of arbitrary length in their model, but demands (in their representation: jobs) are not allowed to be split. From their computational tests they conclude, that their algorithm outperforms those for the DLSP with respect to solution quality and computational times, if either the number of products is small or the test instances exhibit high capacity utilization and significant (i.e. long) setup times.[126]

In another paper, Brüggemann and Jahnke (2000) consider again the DLSP with period overlapping setup times which have to be multiples of a period's length. Furthermore, they introduce the feature of batch availability into their model. In their model batch availability means, that any produced quantities only become available, if production of the specific product has ceased. In our terminology this means, that products only become available, if the campaign is finished. A heuristic based on simulated annealing is presented to cope with these types of problems.[127]

---

122 Cf. Belvaux and Wolsey (2000) pp. 724-738.

123 Cf. among others Salomon (1991) pp. 42-54, Salomon et al. (1991) pp. 805-811, Vanderbeck (1998) pp. 1414-1415, Webster (1999) pp. 768-769 and Brüggemann und Jahnke (2000) pp. 514-517.

124 Cf. Blocher et al. (1999) pp. 559-569.

125 Cf. Cattrysse et al. (1993) pp. 477-478 and Brüggemann and Jahnke (1994) pp. 755-757.

126 Cf. Jordan and Drexl (1998) pp. 698-712.

127 Cf. Brüggemann and Jahnke (2000) pp. 513-514, 517-521.

A CSLP-like problem with additional constraints is tackled by Lee and Chen (2002). In their MIP model formulation minimal and maximal production rates, minimal and maximal storage capacities as well as minimal campaign production quantities are to be respected. Only setup costs, but no setup times are included, but setup states may be carried over for several periods. Minimal campaign production quantities must be met in each period.[128]

Smith-Daniels and Smith-Daniels (1986) also consider a CSLP-like model although several items of one distinct family may be produced in each period, but setup and tear down costs are family dependent only. They present a MIP model formulation with the additional constraints that setup times are necessary between items of a family.[129]

Belvaux and Wolsey (2000) formulate a generic lot-sizing model, which is also valid for small-bucket problems. Their model formulation respects minimal production quantities per period. They propose a specialized branch-and-cut system to deal with these kinds of problems.[130]

In a different paper, Belvaux and Wolsey (2001) present tight mathematical programming model formulations for small bucket problems. These are extended by consideration of minimum campaign quantities. Here, minimum campaign quantities are given either in an integer number of periods or – if this is not the case – it is assumed that within campaigns production must be at full capacity.[131]

The PLSP which has been introduced in the literature by Haase (1994) is presented by the same author with period overlapping setup times of arbitrary length. Although the model formulation is mathematically not correct,[132] the selected solution algorithm remains valid. The solution algorithm is a heuristic approach that moves backward from the last period to the first and decides on the sequencing of products based on a randomized regret measure.[133]

A PLSP embedded in a model formulation of a multi-site production network is presented by Timpe and Kallrath (2000). Their model which is taken from the chemical industry is enhanced by transports and special inventory constraints (tanks). The model formulation is solved by a standard MIP solver.[134]

Kallrath (1999) also introduces several other features into a basic PLSP formulation. In his model campaigns extending over several periods are considered. These can be specified to be in multiples of a distinct batch size or to respect minimal or maximal campaign quantities. Furthermore, he explains how different utilization rates may be included. He proposes a MIP model formulation to solve these kind of problems.[135]

---

128 Cf. Lee and Chen (2002) pp. 16-17, 21-22.

129 Cf. Smith-Daniels and Smith-Daniels (1986) pp. 280-281.

130 Cf. Belvaux and Wolsey (2000) pp. 725-729.

131 Cf. Belvaux and Wolsey (2001) pp. 997-1000.

132 See section 3.4.

133 Cf. Haase (1994) pp. 31-35 and 69-77 and Drexl and Haase (1995) pp. 76-82.

134 Cf. Timpe and Kallrath (2000) pp. 422-435.

135 Cf. Kallrath (1999) pp. 330-337.

4.1.3 Hybrid Models

Of the hybrid models presented in section 2.4, up to now, the CLSPL received most attention in literature. A mathematical model formulation of the CLSPL was presented first in 1979 by Lambrecht and Vanderveken. But at that time they considered the problem as too difficult and therefore dropped the linking constraints and solved a CLSP instead.[136]

Fourteen years later, in 1993, Dillenberger et al. motivated by a production planning problem from practice take up the CLSPL. They present a new MIP model formulation which forms the basis for their solution algorithm. Their algorithm consists of a branch-and-bound scheme, in which the order of binary variables to branch on is given by the period sequence.[137]

Haase (1994) introduces the model's name, CLSPL. He also provides a MIP model formulation, but solves the problem with the help of a stochastic heuristic. The heuristic moves backwards from the last to the first period and applies lot-sizing decisions based on a randomized regret measure.[138]

Later, Haase (1998) changed the model formulation to allow only for one link, thereby forbidding continuous production of one product in three consecutive periods. The reason for this is based on his observation that both models (original CLSPL and modified CLSPL) produce nearly the same solutions, but the modified model seems to be a lot easier to solve.[139]

Gopalakrishnan et al. (1995) propose a model formulation for the CLSPL with product independent setup times. Their model formulation is solved by a standard MIP solver (LINDO). Later, in a technical note, their model formulation is altered to accompany product dependent setup times. Most recently, Gopalakrishnan et al. (2001) apply a meta-heuristic, tabu-search, to the CLSPL.[140]

Sox and Gao (1999) also propose a MIP model formulation for the CLSPL, which has been shown not to be mathematically correct for some cases.[141] Their model formulation is based on a shortest route representation of the problem. A Lagrangian decomposition heuristic is proposed to cope with larger test instances. In line with Haase they experience that restricting the number of consecutive setup carry-overs to one makes the problem much easier without loosing to much from optimality.[142]

---

136 Cf. Lambrecht and Vanderveken (1979) pp. 104-107.

137 Cf. Dillenberger et al. (1993) pp. 108-116, Dillenberger and Wollensak (1994) pp. 93-96 and Dillenberger et al. (1994) pp. 276-283.

138 Cf. Haase (1994) pp. 18-21, 69-79.

139 Cf. Haase (1998) pp. 130-131.

140 Cf. Gopalakrishnan et al. (1995) pp. 1976-1981, Gopalakrishnan (2000) pp. 3421-3423 and Gopalakrishnan et al. (2001) pp. 851-863.

141 Cf. Suerie and Stadtler (2003) pp. 1053-1054.

142 Cf. Sox and Gao (1999) pp. 174-180.

Moreover, Suerie and Stadtler (2003) present a MIP model formulation which is augmented by valid inequalities and a decomposition heuristic. This approach will be reviewed in more detail with some additions in chapter 6.[143]

Porkka et al. (2003) introduce an important feature based on observations from process industries in their version of the CLSPL. In their application, idle time within a campaign is not allowed. Therefore, they add constraints to their model formulation which allow only for continuous production of one product throughout a period with setup carry-overs from the preceding and into the next period, if production is at full capacity in this period.[144]

Recently, Quadt and Kuhn (2003) proposed a (heuristic) solution algorithm for the CLSPL based on a MIP model formulation which allows for backlogging and parallel resources. In their application (from semiconductor industries) the most important characteristic is a huge number of parallel resources which needs to be taken into account. Therefore, they do not model each resource explicitly by binary variables, but introduce integer variables which represent the number of resources that hold a certain setup state. Their heuristic algorithm compares favorably with a MIP model formulation for larger test instances.[145]

The CLSD which is an extension of the CLSPL with sequence dependent setup costs and setup times (see section 2.4.2) is first tackled by Smith-Daniels and Ritzman (1988). They present a MIP model formulation representing a production planning problem which they observed in the food industries.[146]

Heuts el al. (1992) consider a case where production has to be in multiples of a predefined batch size. Their problem statement also includes sequence dependent setup times and maximum storage constraints. In their model formulation batches have to lie completely within a period, but this does not pose a very tight constraint, as they chose demand per product per period to lie between 125 and 275 batches. Two heuristics are proposed to solve this problem.[147]

Later, Haase (1996) considers the CLSD without setup times. He proposes a heuristic which moves from the last to the first period and applies a priority rule for making scheduling decisions. This priority rule is based on two parameters for which a local search method is proposed.[148]

Kang et al. (1999) respect minimal and maximal production quantities. These have to be obeyed for each lot and not for each period, but in their model multiple lots per period are possible and lots cannot extend over several periods. Their solution approach is based on column generation and branch-and-bound.[149]

---

143 Cf. Suerie and Stadtler (2003) pp. 1041-1048.

144 Cf. Porkka et al. (2003) pp. 1136-1139.

145 Cf. Quadt und Kuhn (2003) pp. 185-188 and Quadt and Kuhn (2004).

146 Cf. Smith-Daniels and Ritzman (1988) pp. 651-668.

147 Cf. Heuts et al. (1992) pp. 413-424 and Selen and Heuts (1990) pp. 39-45.

148 Cf. Haase (1996) pp. 51-56.

149 Cf. Kang et al. (1999) pp. 274-280.

Grünert (1998) extends the CLSD to a multi-level product structure. In contrast to Haase's model, Grünert also considers sequence dependent setup times. His solution algorithm is based on Lagrangian decomposition and tabu-search.[150]

Haase and Kimms (2000) propose an algorithm for the CLSD with sequence dependent setup times and sequence dependent setup costs. They assume, that setup operations for a certain product only occur in those periods, in which inventory of this product is empty at the end of the previous period (so-called "zero-switch-property"). However, this is generally not the case in capacitated lot-sizing problems. Their solution approach is a tailor-made branch-and-bound scheme.[151]

A different approach using Lagrangian decomposition to cope with the original CLSD is presented by Magnusson (2001).[152]

Clark and Clark (2000) present a MIP model formulation for the CLSD with backlogging, but they consider only inventory and backlog costs and no setup costs in the objective function. Their solution algorithm is based on their MIP model formulation which is combined with an LP model for periods which belong to subsequent planning cycles (based on the assumption of a rolling horizon).[153]

Recently, Timpe (2002) proposes an algorithm for the CLSD which combines MIP and constraint programming. The constraint programming part is used to derive schedules within periods, while the products to be produced in each period are determined by a MIP model, which acts as a master process in this approach. The underlying problem is taken from industry and inherits minimal and maximal production quantities for each lot, but these conditions have to be fulfilled for each period independently.[154]

As we have seen in section 2.4.3, the GLSP is the most general hybrid model. It has been introduced first by Fleischmann and Meyr (1997) who present a heuristic solution algorithm based on threshold accepting. In their paper, they consider two variants of the GLSP, one that loses setup states after idle time and one that preserves setup states. Both variants include restrictions on the minimum production quantity for each lot. These minimum required quantities have to lie completely within one period.[155]

More variants of the GLSP are provided by Meyr (1999, 2000, 2002) and solved with a heuristic combining meta-heuristics (threshold accepting, simulated annealing) with dual reoptimization.[156]

Furthermore, Meyr (2004) provides a MIP model formulation for a multi-level GLSP. Here, emphasis is on the coordination of different production lines. Solutions to this model formulation are generated by a standard MIP solver (CPLEX), but Meyr (2004) acknowledges that this is only appropriate for small examples.[157]

---

150 Cf. Grünert (1998) pp. 47-56 and 97-144.

151 Cf. Haase and Kimms (2000) pp. 161-165.

152 Cf. Magnusson (2001) pp. 6-14 and 36-41.

153 Cf. Clark and Clark (2000) pp. 2290-2295.

154 Cf. Timpe (2002) pp. 435-445.

155 Cf. Fleischmann and Meyr (1997) pp. 12-18.

156 Cf. Meyr (1999) pp. 75-201, Meyr (2000) pp. 313-319 and Meyr (2002) pp. 279-286.

157 Cf. Meyr (2004) pp. 589-607.

Stammen-Hegener (2002) also considers a multi-level variant of the GLSP. She does not provide any solution algorithm, but a mere model formulation. Her emphasis is on reducing abundance in the time structure and on the exact modeling of holding costs. Unfortunately her model formulations become non-linear.[158]

A model formulation which is related to the multi-level GLSP is presented by Kimms and Motta Toledo (2003). They present an interesting MIP model formulation which is focusing on a practical application and therefore devote much effort to that case. They introduce two time domains. The first one is a uniform time discretization which they call micro periods and which is used to align the two stages of their production planning problem. The second one is based on slots with one slot for each lot. This time domain is used to enforce e.g. minimal campaign quantity restrictions.[159]

4.2 Models Originating from Process Industries

Models for production planning designated to process industries often contain problem specific characteristics and therefore several representations have been developed to describe these problems with their specific characteristics thoroughly. The most relevant of these characteristics and problem representations are introduced in the following subsection (4.2.1). Pursuing these a brief literature review is given focusing on the more recent publications in this area.[160] The literature review is divided into two parts. The first part (4.2.2) reviews models with a discrete, i.e. bucket-oriented, time structure, whereas the second part (4.2.3) reviews models based on a continuous time scale. The reason for this is twofold. First, the models proposed in this thesis are bucket-oriented and therefore closer related to the discrete time models reviewed first. On the other hand, the models proposed here can generally represent all solutions that are possible within a continuous time setting. Therefore, these models might as well be compared to continuous time models. Second, a separated review is justified because of the objective pursued here. Although most often time-based objectives (e.g., minimization of makespan) are in the focus of (short-term) models for production planning in the process industries, here a cost-based approach is followed, because the application of these models is intended in medium-term planning. This is not easily done in models based on a continuous time scale, because e.g. the problem of calculating inventory holding costs becomes non-linear as soon as the time interval between two successive events becomes variable.

---

158 Cf. Stammen-Hegener (2002) pp. 126-208.

159 Cf. Kimms and Motta Toledo (2003) pp. 2-14.

160 A careful literature review on production planning in the process industries (before 1998) can be found in Blömer (1999) pp. 38-49, 61-63, 73-85 and Pinto and Grossmann (1998) pp. 438-457.

4.2.1 Characteristics and Representation of Models from Process Industries

Production planning approaches in the process industries vary from those in manufacturing mainly due to the different nature of the production process.[161] More differences between these two types of industries are presented in Table 4-1 taken from Ashayeri et al. (1995).[162] In the following the focus will be on the aspect of modeling. In this area several more distinctive elements come into mind.[163]

First, in the process industries products or intermediates are far more often perishable. This means that stocking policies have to be obeyed, when planning for production.[164] Usually products and intermediates are therefore classified into three categories according to their perishability: zero-wait, unlimited-wait and finite-wait. Steady processing of zero-wait products must be arranged for during planning.

Second, coupled with perishability is the question of stocking. Storage capacity is often scarce in the process industries, because special equipment is needed and this equipment is often dedicated to certain products. For example, if tanks are needed as storage equipment, only one product can use each tank at each point in time and often cleaning operations are required, if the tank needs to be used by another product.

Third, processing times might require special modeling. Whereas in discrete manufacturing processing times for a certain lot are usually dependent on the lot size, i.e. the number of units to be produced, this is often not true in the process industries. Here, processing times are often constant, irrespective whether the reactor is filled to 70% or 90% of its capacity.[165]

Finally, time consuming and costly setups can be identified as characteristic to process industries.[166] This aspect has already been highlighted in chapter 3, where the advantages of realistic modeling of setup operations and setup states has been discussed.

Based on these characteristics, various representations of the production process have been developed which differ from those known from discrete manufacturing. There, the concept of parts lists, bill of materials (BOM) or the extension to product/operation structures prevail. BOMs consist of lists of components,

---

161 E.g., Applequist et al. (1997) pp. 87-89. For a detailed analysis regarding the differences of production systems within process industries see Dennis and Meredith (2000) pp. 1091-1096.

162 More elaborate characterizations of especially the process industries can also be found in Loos (1997) pp. 17-66, Blömer (1999) pp. 5-36 and Kießwetter (1999) pp. 7-37.

163 E.g., Pinto and Grossmann (1998) pp. 434-438.

164 Cf. Crama et al. (2001) p. 3 and Pinto and Grossmann (1998) p. 437.

165 Cf. Dessouky and Kijowski (1997) p. 399. Reactors are sometimes not filled completely, because they may have a higher yield if filled only to a certain amount or because there is not enough demand for the generated product. Cf. Blömer (1999) p. 25.

166 Cf. Shapiro (1993) p. 397.

which are needed to manufacture a certain product and show the requirement of each item for each item. They are usually organized in a tree structure with the end product at the top (see Fig. 2-1, p. 12). In contrast to parts lists, BOMs usually show how products are assembled.[167] Product/operation structures depict not materials, but operations as the smallest unit that is planned for. Furthermore product/operation structures tie the operations to resources.[168]

The most basic and widespread concept taking the role of BOMs in the process industries are recipes.[169] Recipes contain all information necessary to produce a certain product. These are lists of ingredients, descriptions of processes, information on necessary resources, description of products and by-products as well as information on hazardous by-products and intermediates that might result during the production process.[170] A very important feature of recipes is that they often allow for alternative ways to generate a certain product and that variable yields can result.[171] Although recipes contain all data relevant for planning, they are often not used as sole source of information, because the information is often not presented clearly, as recipes hold much more data not relevant for planning.[172]

Consequently, in the chemical engineering literature different concepts resuming the role of BOMs in the process industries have been developed. In the following three of them will be introduced briefly, because almost all models discussed in the two subsequent sections build on either of them. The three concepts are the state-task network (STN), the resource-task network (RTN) and the state-sequence network (SSN). To illustrate the differences between these approaches, an example will be presented first, which is then modeled using each approach.

In the example[173] two products ("1" and "2") are made of three commodities ("A", "B" and "C") (see Fig. 4-1 for an STN, Fig. 4-2 for an RTN and Fig. 4-3 for an SSN representation). The production process requires several operations / processes. First, "A" needs to be heated for one hour, yielding "Hot A". Furthermore, a chemical reaction lasting two hours and requiring equal amounts of commodities "B" and "C" yields intermediate "BC". A second chemical reaction requiring two fractions of "Hot A" and three fractions of "BC" lasts two hours and yields two fractions of "1" and three fractions of intermediate "ABC". The third reaction uses one fraction of "C" and four fractions of "ABC" and requires one hour to yield "ABCC". Last, "ABCC" is distilled yielding nine fractions of "2" (after one hour) and one fraction of "AB" (after two hours). Four resources can be utilized in this process: a heater (capacity: 100kg) for heating of "A", two reactors capable of performing the three reactions mentioned (capacities: 80kg and 50kg) and one distiller (capacity: 200kg). Unlimited storage is available for the products and com-

167 Cf. Hill (2003) p. 4.

168 Cf. Tempelmeier and Helber (1994) pp. 297-298 and Tempelmeier and Derstroff (1996) p. 739.

169 Cf. Crama et al. (2001) p. 13.

170 Cf. Loos (1997) pp. 174-175.

171 Cf. Crama et al. (2001) pp. 14-15.

172 Cf. Blömer (1999) p. 28.

173 Cf. Kondili et al. (1993) pp. 215-216.

modities, but is limited for intermediates ("Hot A", "ABCC": 100kg each; "BC": 150kg and "AB": 200kg).[174]

Table 4-1: Differences between process industries and discrete industries.[175]

	Process industries	Discrete industries
Relationship with the market		
Product type	Commodity	Custom
Product assortment	Narrow	Broad
Demand per product	High	Low
Cost per product	Low	High
Order winners	Price, Delivery Guarantee	Speed of delivery, Product features
Transporting costs	High	Low
New products	Few	Many
The production process		
Routings	Fixed	Variable
Lay-out	By product	By function
Flexibility	Low	High
Production equipment	Specialized	Universal
Labor intensity	Low	High
Capital intensity	High	Low
Changeover times	High	Low
Work in process	Low	High
Volumes	High	Low
Quality		
Environmental demands	Yes	Hardly
Danger	Sometimes	Almost never
Quality measurement	Sometimes long	Short
Planning & Control		
Production	To stock	To order
Long term planning	Capacity	Product design
Short term planning	Utilization capacity	Utilization personnel
Starting point planning	Availability capacity	Availability material
Material flow	Divergent + convergent	Convergent
Yield variability	Sometimes high	Mostly low
'Explosion' via	Recipes	Bill of material
By and Co-products	Sometimes	Not
Lot tracing	Mostly necessary	Mostly not necessary

The STN has been introduced first by Kondili et al. (1993). It is based on two types of nodes which are connected via directed arcs. The first type (state nodes, depicted by circles in Fig. 4-1) represents products and intermediates, whereas the second type (task nodes, depicted by boxes in Fig. 4-1) represents operations and

[174] Cf. Kondili et al. (1993) pp. 215-216.

[175] Ashayeri et al. (1995) p. 3.

processes that transform the states from an input state to an output state. STNs consider primarily the chemical structure of the production process, but not the physical structure (assignment of tasks to resources). If parallel resources prevail, i.e. tasks may be assigned to more than one resource, this can be indicated by grouping tasks in the STN representation or by addition of a second graph (resource graph).[176] It is important to note for planning processes that STNs do not need to be connected graphs. A disconnected graph results if two products consist of disjoint sets of intermediates, but share one resource in at least one processing step. Then, when planning for the joint resource (both) STNs need to be taken into account.[177]

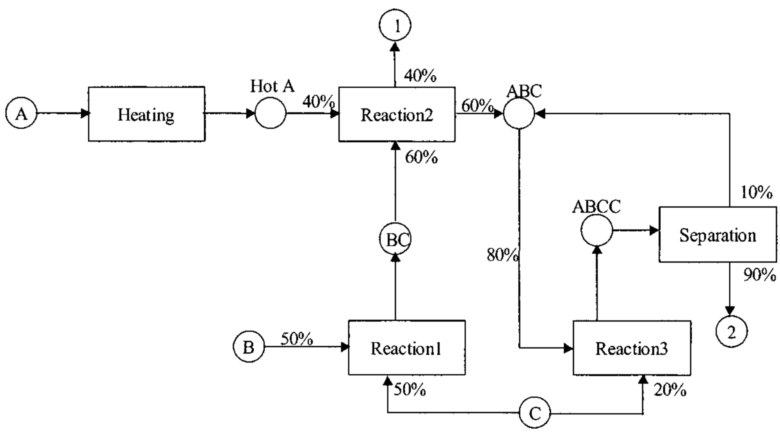


Fig. 4-1: STN representation of example (adapted from Kondili et al. (1993) p. 215).

One disadvantage of the STN representation is the non-existent coupling of tasks and resources, if tasks can be assigned to multiple resources. Furthermore, it has been criticized that tasks are always coupled with material transformation. Often cleaning processes are process steps which may not be neglected, but as these tasks are usually related to resources and not to materials they cannot be modeled easily.[178]

The RTN representation[179] overcomes some of the aforementioned disadvantages. Similar to the STN it is based on two types of nodes. The first type of nodes (resource nodes, depicted in Fig. 4-2 as circles) represents resources, whereas the second type of nodes (task nodes, depicted in Fig. 4-2 as boxes) represents tasks. The concept of tasks in RTNs is more general than in STNs. Tasks may be any processing steps that transform one set of resources into another set of resources. Thereby, tasks may in addition to operations denote transportation or storage

176 E.g., Blömer (1999) p. 31.
177 Cf. Kondili et al. (1993) pp. 213-215.
178 Cf. Blömer (1999) p. 31.
179 Cf. Pantelides (1994) pp. 267-272.

processes. Furthermore, in this context resources may not only stand for materials (states in the STN representation), but also correspond to energy, manpower, storage and transportation facilities. Fig. 4-2 shows the RTN representation of the example. Dotted lines indicate, that some resources (e.g., reactor 1 and 2) are input as well as output of a certain task.

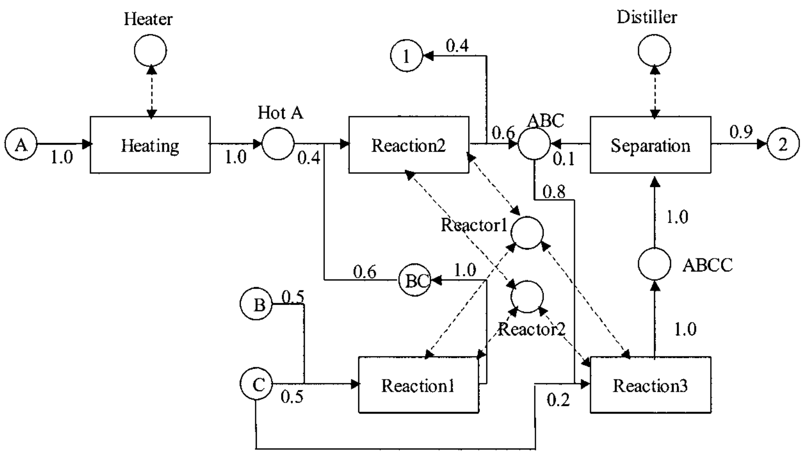


Fig. 4-2: RTN representation of example (adapted from Castro et al. (2001a) p. 2063).

RTNs provide the missing link between resources and tasks of STNs. Cleaning operations are easily modeled, e.g. by a task which consumes a dirty resource, cleaning materials and workforce and turns out the cleaned resource, the workforce and some waste.

The most current process representation, SSNs,[180] again build on STNs. In contrast to these, there is only one type of nodes present (state nodes, depicted as circles in Fig. 4-3). These state nodes directly correspond to the state nodes in STNs, i.e. products, intermediates and commodities. Tasks, i.e. processes/operations, are only given implicitly by the sequence of the states. If a state changes, a task must have been performed, i.e. the sequence of states corresponds to tasks. Thereby, three building blocks of SSNs can be distinguished. The first one is a state transformation indicated by an arc connecting two states (e.g., "A" → "Hot A" in Fig. 4-3), the second one is a reaction indicated by two or more states forming a new state (e.g., "B", "C" → "BC" in Fig. 4-3), and the third one is indicated by splitting one state into two or more states (e.g., "ABCC" → "2", "ABC" in Fig. 4-3). The main reason for getting rid of the task nodes in the SSN representation is that binary variables that are needed to model tasks in formulations using the STN representation can be saved, thereby yielding tighter model formulations.[181]

[180] Cf. Majozi and Zhu (2001) pp. 5936-5938.

[181] Cf. Majozi and Zhu (2001) p. 5936.

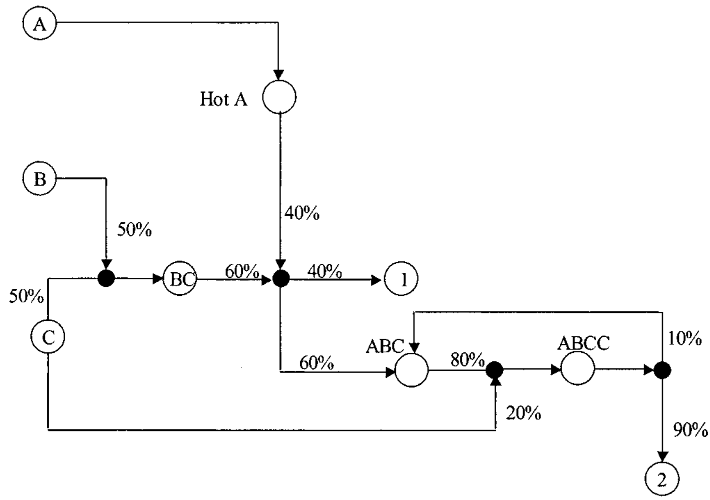


Fig. 4-3: SSN representation of example (adapted from Majozi and Zhu (2001) p. 5943).

Although a first abstraction, STNs are closest related to recipes among the presented representations. Their main disadvantage – the missing link between tasks and resources – is overcome by RTNs. Furthermore, in RTNs the concept of states is extended to comprise all types of resources. On the other hand, the inclusion of resources leads to a slight loss in clarity, if certain resources can handle a lot of tasks. SSNs resemble STNs without tasks and have been developed for modeling purposes to decrease the number of binary variables necessary. All of these representations have been used to tackle various production planning problems from process industries. The resulting model formulations will be discussed in the following two subsections based on their approach how to model time.

4.2.2 Discrete Time Model Formulations

Comprehensive literature reviews on model formulations for process industry problems have already been conducted.[182] Because of the vast amount of literature published in this area, in the following the focus will be on those published since 1998 and therefore include only a selection of the most important contributions prior to 1998. As the terminology used in the literature often varies between different references, the following terminology relating to the representation of time is introduced: A model formulation is called a discrete time model formulation, if the sequence of the periods and the length(s) of the periods are parameters. This is in contrast to a continuous time model formulation in which the duration of periods and/or their number are not known a priori. A discrete time model formulation

[182] E.g., Shah (1998) pp. 78-83 and Blömer (1999) pp. 73-85.

is called uniform, if all periods are of equal length, and non-uniform, if periods of different length are present. If the same time grid is imposed on the complete problem, this will be called a common time grid. In contrast, a non-common time grid prevails, if e.g. different resources operate on different time scales.

The seminal paper by Kondili et al. (1993) introduced the STN representation. A MIP model formulation is presented to solve the arising scheduling problem. The model formulation is based on time discretization with uniform time buckets (Table 4-2). Binary variables are used to model if a resource m starts a task i at the beginning of period t. As the start or end of a task is only allowed at period boundaries, a common time grid is induced for all resources, which is an advantage as well as a drawback of the model. On the one hand, it allows to position each task against all competing tasks on one reference grid, while on the other hand, this leads to a large number of binary variables, especially if processing times of different tasks vary, because the period length has to be chosen as the greatest common factor of all process lengths. Thereby, in a sense the model formulation shows similarities to the DLSP, where changes in production are also allowed only at period boundaries and within periods full production is assumed (all-or-nothing assumption). Furthermore, problem specific constraints like temporary unavailability of resources or limited availability of utilities are modeled. The authors admit that the model gets too large for practical problems, but the contribution of this paper remains the introduction of the STN representation.

A problem from pesticide manufacturing is tackled by Dessouky and Kijowski (1997). There, the objective is to minimize inventory holding costs, overtime costs and a cost component which depends on the fixed batch size. This last cost component stems from configuring the plant to a batch size which is then assumed fixed during the planning interval. As this cost component is non-linear in the batch size, a MINLP model formulation results. Due to assumptions that are valid in the case studied by them, they are able to develop an optimizing procedure.[183]

Mockus and Reklaitis (1997) present a MINLP model formulation based on the STN representation. Their initial model formulation uses a continuous time representation, but is simplified to reduce non-linearities to a non-uniform time discretization. The authors claim that the proposed non-uniform model formulation matches the model formulation by Kondili et al. (1993), if a uniform time discretization is applied, but is more general, as it is not dependent on the equal period length which has to be chosen there.[184]

A MIP model formulation intended for mid-term planning respecting minimal campaign quantities is proposed by McDonald and Karimi (1997). The model formulation relies on a uniform time discretization, but does not consider setup or cleaning times explicitly. Minimal campaign quantities have to be fulfilled within one or two successive periods.[185]

This restriction is lifted in a companion paper intended for short-term scheduling. There one model formulation based on a continuous time scale and one based

183 Cf. Dessouky and Kijowski (1997) pp. 399-408.

184 Cf. Mockus and Reklaitis (1997) p. 1153 and Kondili et al. (1993) pp. 216-221.

185 Cf. McDonald and Karimi (1997) pp. 2691-2700.

on non-uniform time discretization is proposed. In the latter, time periods are defined by due dates for product deliveries of the company. Similar to the GLSP (see section 2.4.3) an arbitrary number of time slots of arbitrary length to be determined by the solution algorithm is assigned to each time period. The time periods (i.e. due dates) supply a common time grid across all resources, whereas the lengths of the individual time slots are not synchronized.[186]

The same representation of time is used in a later paper by Lim and Karimi (2003b). As the individual time slots are not synchronized, resources used by many tasks in parallel cannot be modeled. To overcome this shortcoming so-called checkpoints are identified, which correspond to the beginning of each task. Resource availability is then checked at these checkpoints. Unfortunately this leads to a tremendous increase of the number of variables and constraints of the model formulation.[187]

An RTN-based MIP model formulation working on a uniformly discretized time axis is proposed by Yee and Shah (1998). Their aim is to reduce the large observed integrality gaps of such a formulation. Their analysis suggests that this integrality gap stems to a large extent from the modeling of changeovers (setups). Therefore, they introduce additional constraints ensuring that at least a minimum – a priori determined – number of changeovers is planned for. Furthermore, they apply a standard reformulation technique known from lot sizing (SPL reformulation[188]) to their model formulation. On their test bed, both measures proved to be quite effective.[189]

Blömer and Günther (1998) use the STN representation to develop a MIP model formulation. In their problem batch processing times are independent of the batch size, sequence dependent cleaning times are respected and constraints to model special storage policies (e.g., zero-wait) are included. The objective chosen is to minimize makespan. Making use of uniform time discretization, their model formulation is intractable for commercial MIP solvers. Therefore, different LP-based heuristics are employed. One of them decomposes the problem at hand into single-level problems, whereas another covers the time grid with a pattern of allowed start times for processes, thereby reducing the number of binary variables indicating the start of a process in a certain period significantly. According to their computational tests the time grid based approach performed best.[190]

Later, the solution approach has been further refined as well as has the applicability of the model formulation broadened by Blömer (1999) and Blömer and Günther (2000).[191]

The solution approach proposed by Rodrigues et al. (2000) for the STN-based MIP model by Kondili et al. (1993)[192] builds on time decomposition. Based on the

186 Cf. Karimi and McDonald (1997) pp. 2701-2714.
187 Cf. Lim and Karimi (2003b) pp. 6832-6842.
188 Cf. Rosling (1986) p. 121 and Stadtler (1996) pp. 570-571.
189 Cf. Yee and Shah (1998) pp. S403-S410.
190 Cf. Blömer and Günther (1998) pp. 245-259.
191 Cf. Blömer (1999) pp. 93-126 and Blömer and Günther (2000).
192 Cf. Kondili et al. (1993) pp. 216-221.

problem data time windows are derived for the production of each batch. These time windows are further reduced by taking into account capacity considerations with the help of a constraint-propagation mechanism. When this is done, binary variables are introduced only within the time windows resulting in a significantly smaller problem size than the original model.[193]

Trautmann (2001) devises a solution algorithm for a so-called batch-scheduling problem. His approach requires the batch size to be fixed in a prior planning step. His aim is then to schedule the batches taking into account aspects of perishability, sequence dependent setup times, availability of manpower, storage constraints and work break calendars. He proposes a heuristic based on branch-and-bound to obtain solutions to scheduling problems of practically relevant size.[194]

Lee et al. (2002) propose a MIP model formulation based on non-uniform time discretization. The planning horizon is divided into shorter periods in the near future and larger periods towards the end of the planning horizon (telescopic time scale). Their model constrains minimum campaign quantities with no restriction regarding the number of periods within which this restriction has to be fulfilled. Thereby, campaigns spanning over several periods are allowed. Furthermore, planned outages (e.g., for maintenance) are respected. Backorders are allowed in the model and safety stock violations are penalized. In addition to these two cost components, the objective function minimizes (sequence dependent) setup and inventory holding costs. Regarding the model size (number of constraints, binary and continuous variables), the model formulation compares favorably to those of Karimi and McDonald (1997) and Ierapetritou et al. (1999).[195]

A new representation of time is introduced by Lim and Karimi (2003a). First, the planning horizon is discretized according to due dates of orders. Furthermore, a number of time slots is associated with each due date. In contrast to the time representation by Karimi and McDonald (1997)[196], who divide the time between two due dates into a predefined number of slots, here the assignment is such, that the first K_1 slots fill orders for the first due date, the first K_2 ($\geq K_1$) slots fill orders for the second due date and so on. At the same time, slots associated with a due date need not to be finished before the corresponding due date, thus allowing tardiness (backorders). Their model respects batch size limitations (minimum and maximum) with batch processing times consisting of a fixed and variable part. Furthermore, (sequence dependent) setup times are modeled. Their computational results reveal that solution times are significantly influenced (in an erratic manner) by the selection of M in so-called big-M constraints.[197] Compared to other MIP

193 Cf. Rodrigues et al. (2000) pp. 3823-3834.

194 Cf. Trautmann (2001) pp. 5-100.

195 Cf. Lee et al. (2002) pp. 58-66.

196 Cf. Karimi and McDonald (1997) p. 2703.

197 Big-M constraints are constraints taking the form $x \leq M{\cdot}y$ with a continuous variable x and a binary variable y and M chosen such that it is not limiting the feasible solution space.

model formulations from literature they conclude, that their model is more robust with respect to the selection of M.[198]

Castro et al. (2003) consider the problem to obtain a cyclic schedule with maximum throughput. Their model formulation is based on the RTN representation and uniform time discretization. In a case study they compare their discrete time MIP model formulation with one based on continuous time representation. Although they make use of a very short period length (five minutes), the model formulation based on time discretization outperforms their benchmark.[199]

Maravelias and Grossmann (2003b) present an algorithm based on successive solution of MIP model formulations to determine the optimal makespan for a short-term scheduling problem. Their model formulation utilizes the STN representation and uniform time discretization. Batch sizes may vary between certain limits (minimum and maximum), but these have to be obeyed for each period independently.[200]

Table 4-2 summarizes the literature review on discrete time model formulations. For each reference it is indicated which process representation has been used, whether the process is characterized by some sort of batching considerations, how time is modeled and which solution method has been suggested by the authors.

Table 4-2: Classification of literature (discrete time model formulations).

Author(s)	Process Representation	Process characteristic	Modeling of time	Solution method
Blömer (1999)	STN	batch	uniform	LP-based heuristic
Blömer and Günther (1998)	STN	batch	uniform	LP-based heuristic
Blömer and Günther (2000)	STN	batch	uniform	LP-based heuristic
Castro et al. (2003)	RTN	-	uniform	MIP, algorithm
Dessouky and Kijowski (1997)	-	batch	uniform	MINLP, algorithm
Karimi and McDonald (1997)	-	-	non-uni.	MIP
Kondili et al. (1993)	STN	batch	uniform	MIP
Lee et al. (2002)	-	cont.	non-uni.	MIP
Lim and Karimi (2003a)	-	batch	non-uni.	MIP
Lim and Karimi (2003b)	-	-	non-uni.	MIP
McDonald and Karimi (1997)	-	-	uniform	MIP
Maravelias and Grossmann (2003b)	STN	batch	uniform	MIP, algorithm
Mockus and Reklaitis (1997)	STN	batch	non-uni.	MINLP, MIP
Rodrigues et al. (2000)	STN	batch	uniform	MIP, heuristic, CP
Trautmann (2001)	-	batch	(uniform)	heuristic
Yee and Shah (1998)	RTN	-	uniform	MIP, cuts

Although these models based on time discretization have been utilized to solve problems from industry, they contain several disadvantages:[201] First, to capture

198 Cf. Lim and Karimi (2003a) pp. 1914-1924.
199 Cf. Castro et al. (2003) pp. 3346-3360.
200 Cf. Maravelias and Grossmann (2003b) pp. 6252-6257.
201 Cf. Shah (1998) p. 83.

enough detail, a fine resolution of time is needed, resulting in large MIP model formulations. Otherwise, processing times need to be rounded, which introduces some margin of error into the model that might jeopardize the optimality or even the feasibility of solutions. Second, continuous and semi-continuous processes have to be approximated. Third, processing times which depend on the batch size are difficult to model and therefore are considered seldom. Only Lee et al. (2002) and Lim and Karimi (2003a) consider variable processing times in their model formulation.[202] Last, minimum campaign quantities can only be accounted for by complicated constraints, which has been done by McDonald and Karimi (1997), Lee et al. (2002) and Lim and Karimi (2003b).[203]

This is why models based on a continuous representation of time have been developed, which will be discussed in the next section.

4.2.3 Continuous Time Model Formulations

In continuous time model formulations also assumptions based on the representation of time need to be made. Three different representations of time will be distinguished (see Fig. 4-4).[204] All of them have in common, that the period length is unknown at the time of planning and the duration of periods is usually not equal. The first time representation (cont1) is characterized by the fact, that each task (operation) has to start and end at a certain time point (period boundary), whereas in the second time representation (cont2) this constraint is somewhat relaxed and only the beginning (or only the end) of tasks needs to interfere with time points. Both time representations usually rely on a common time grid for all resources. This is no longer true for the third possibility to represent time, which will be called "event-based" in the following (event). There, the planning horizon is divided into a number of events. Events on different resources are coupled by sequencing constraints, but the third event on one resource may well be finished prior to the second event on another resource. The event-based representation of time leads to even smaller model sizes with respect to the number of (binary) variables than the two other approaches to model time continuously.

An RTN-based model formulation is presented by Zhang and Sargent (1996). It relies on a common time grid for all resources. Batch processing times can be variable, but this leads to a MINLP model formulation. The model formulation is linearized using the assumption that processing times are fixed. The authors acknowledge that there exists no algorithm for solving their MINLP model formulation, but indicate that their solution algorithm seems promising, if some assumptions regarding the planning problem are met.[205]

---

202 Cf. Lee et al. (2002) pp. 60-61 and Lim and Karimi (2003a) p. 1916.

203 Cf. McDonald and Karimi (1997) pp. 2695-2696, Lee et al. (2002) p. 61 and Lim and Karimi (2003b) pp. 6834-6836.

204 E.g., Maravelias and Grossmann (2003a) p. 3057.

205 Cf. Zhang and Sargent (1996) pp. 897-904. Some extensions and improvements are presented in Xueya and Sargent (1998).

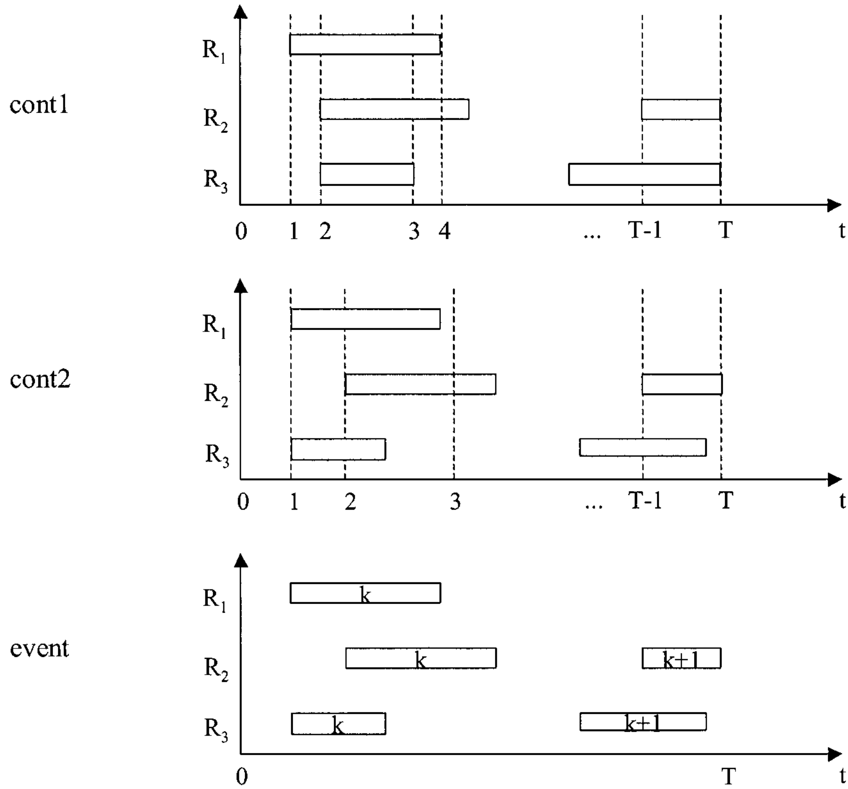


Fig. 4-4: Alternative representations of time (adapted from Maravelias and Grossmann (2003a) p. 3057).

The MIP model formulation by Schilling and Pantelides (1996) is also based on the RTN representation. Tasks can only interact at period boundaries following the time representation denoted by cont1 in Fig. 4-4. Their non-linear model is linearized and then solved by a branch-and-bound algorithm. The algorithm is non-standard, because it branches not only on binary variables, but also on continuous variables (the lengths of the periods). Although their results are encouraging, they state that the computational burden for model formulations based on a continuous time representation remains high.[206]

A model which incorporates the DLSP[207] as a special case is proposed by Jordan (1996).[208] In his model, orders are already assigned to batches and only the sequencing of batches is in the scope of his approach. Batches are not allowed to

[206] Schilling and Pantelides (1996) p. S1226.

[207] See also sections 2.3.1 and 4.1.2.

[208] Cf. Jordan (1996) pp. 100-106.

be split, resulting in an a priori known (and constant) processing time for each batch. Several heuristics are proposed.[209]

Karimi and McDonald (1997) propose an event-based model formulation, although their model relies also on a second time grid based on non-uniform time discretization. The planning interval is divided into a number of time slots which are not synchronized across resources. Timing of tasks and, e.g., the observation of minimal campaign lengths, is based on the continuous time scale (time slots), whereas the flow of material (inventory balance constraints, fulfillment of demands) is done on the discrete time scale. They conclude from their computational tests that their discrete time model formulation with a time representation similar to the GLSP (see also section 4.2.2) outperforms their continuous time model formulation.[210]

Ierapetritou and Floudas (1998a) use the STN representation throughout their paper. Their MIP model formulation differs from the model formulation introduced so far in that it decouples tasks from resources. Instead of defining tri-indexed binary variables indicating that a task j starts at resource m at the beginning of period t, two sets of bi-indexed variables are defined, one relating tasks and periods and the other one relating resources and periods. Sequencing constraints ensure the fit of these two sets of variables. Consequently, the time representation has to rely on a common time grid for all resources/tasks. Especially the reduction in binary variables in the model formulation leads to a significant performance improvement compared to earlier approaches.[211]

In a companion paper the same authors extend their model formulation to take semicontinuous and continuous processes into account. The difference resides in the fact that in continuous processes, intermediates become available continuously opposed to batch processes, where intermediates become available only if the complete batch has been processed (at the end of each period). This requires several modifications to the sequencing constraints.[212]

In a third paper Ierapetritou et al. (1999) consider intermediate due dates in contrast to the former two papers, in which the complete demand has to be met at the end of the planning interval. Their MIP model formulation is altered accordingly. Furthermore, the integration of constraints forcing the observation of minimal campaign quantities is shown, but only for a special case that allows to fulfill this minimal campaign quantity requirement in two consecutive periods.[213] Later, Lin et al. (2002) show, based on a case study, how this short-term scheduling solution approach is integrated into a hierarchical planning framework based on the decomposition of time into independent planning intervals.[214]

---

209 Cf. Jordan (1996) pp. 10-147.

210 Cf. Karimi and McDonald (1997) pp. 2701-2714.

211 Cf. Ierapetritou and Floudas (1998a) pp. 4341-4359.

212 Ierapetritou and Floudas (1998b). Giannelos and Georgiadis claim that storage requirements are underestimated in this model formulation and provide a counter-example. Cf. Giannelos and Georgiadis (2002b) pp. 2435-2436.

213 Cf. Ierapetritou et al. (1999) pp. 3446-3461.

214 Cf. Lin et al. (2002) pp. 3884-3906.

Mockus and Reklaitis (1999a, 1999b) use the STN representation to propose a MINLP model formulation. Their model formulation is based on the assumption that each task can only be performed once on a resource within the planning interval. Thereby, the need arises to define several identical tasks, if operations need to be performed several times (e.g., several batches need to be processed). The representation of time in their model formulation corresponds to cont1. As a solution method a so-called Bayesian heuristic approach is chosen. The central idea is to perform a sequence of heuristic rules/decisions (e.g., the choice of the next object in a knapsack problem) based on some probability and doing this iteratively.[215]

A continuous time model formulation based on the RTN representation is proposed by Castro et al. (2001a). Time is modeled similar to cont2, but shows also characteristics of cont1. There is a common time grid for all resources and all tasks are coupled to a beginning and an end time point. However, the duration of a task does not have to match the difference between these two time points, but it can also be smaller. This avoids non-linearity in the model formulation. Furthermore, the start and end time point do not have to be consecutive time points, thereby allowing basically batch sizes of arbitrary size, which are allowed to have variable processing times.[216]

In a later paper, Castro et al. (2004) improve the model formulation by changing several constraints which yields a better linear relaxation. Furthermore, the applicability of the MIP model formulation is extended to cover also continuous processes.[217]

Lee et al. (2001) devise a MIP model formulation based on the STN representation which becomes non-linear if sequence dependent setup times need to be integrated. Time representation is based on cont1 with a common time grid for all resources. Three tri-indexed binary variables are used to couple tasks, resources and periods. The first one marks the beginning of a task on a resource at a certain time point, the second one shows the processing of a task on a resource at a certain time point and the third one marks the end of a task on a resource at a certain time point. Consequently, tasks can last arbitrarily long. As in most continuous time model formulations, variable processing times can be accounted for easily.[218]

The SSN representation has been proposed by Majozi and Zhu (2001) to further reduce the use of binary variables compared to former MIP model formulations. In their approach only bi-indexed binary variables are necessary signaling that a certain state is used at a certain time point. This is due to their new representation of the production process. Another point worth mentioning is that in contrast to all other approaches published so far, they allow processing times, which do not only

215 Cf. Mockus and Reklaitis (1999a) pp. 197-203 and Mockus and Reklaitis (1999b) pp. 204-210.

216 Cf. Castro et al. (2001a) pp. 2059-2068. Their criticism to other MIP model formulations from literature is mainly based on a slightly different data set used by other authors. This is acknowledged by the authors in a later note. Cf. Ierapetritou and Floudas (2001) pp. 5040-5041 and Castro et al. (2001b) p. 5042.

217 Cf. Castro et al. (2004) pp. 105-118.

218 Cf. Lee et al. (2001) pp. 4902-4911.

vary according to the batch size (material processed), but also according to various other factors (e.g., catalyst types, raw material purity).[219]

Hui and Gupta (2001) avoid the use of tri-indexed binary variables similar to the approach of Ierapetritou and Floudas (1998a)[220] by defining three sets of bi-indexed binary variables. One accounts for the sequence of orders, the second accounts for the assignment of an order to a resource and the third one indicates which order is assigned first to a specific resource. In contrast to the other continuous time model formulations the authors assume constant batch sizes. Furthermore, they devise a heuristic to obtain the sequence of orders to be processed on a certain resource prior to starting their MIP model formulation. The use of this pre-ordering heuristic significantly reduces the number of binary variables and therefore speeds up the solution process. Anyhow, the optimal solution may not be found with this approach.[221]

A model formulation based on time representation cont1 is proposed by Lamba and Karimi (2002a). They also consider minimal campaign quantities in their model formulation. Furthermore, several additional resource constraints are discussed (limitations on the consumption of a common resource, limitations on simultaneous production of a certain product etc.). As MIP model formulations grow too big for real-world problems, they first try different solver options (CPLEX options on branching and simplex pricing) to improve computational performance (with great success) and then propose a two-step decomposition heuristic in a companion paper.[222]

The MIP model formulation by Giannelos and Georgiadis (2002a) uses an event-based representation of time. Binary variables are only bi-indexed, indicating whether a certain task ends at an event point or not. The assignment of tasks to resources by explicit binary variables is avoided by defining multiple tasks, one for each resource. The correct material flow is guaranteed by defining a common event grid for those tasks that produce (or consume) the same states by special sequencing constraints. Tasks can have variable processing times (dependent on the amount of material processed), but need not to start or end precisely at event points. Due to the event-based representation of time, their MIP model formulation requires less binary variables than earlier model formulations based on their evaluation of models from literature.[223]

In a companion paper the same approach is used for continuous processes.[224]

Wang and Guignard (2002) present a MIP model formulation with a representation of time similar to cont2. A common time grid is used for all resources. They do not compare their approach to other model formulations computationally.[225]

---

219 Cf. Majozi and Zhu (2001) pp. 5935-5949.

220 Cf. Ierapetritou and Floudas (1998a) pp. 4341-4359.

221 Cf. Hui and Gupta (2001) pp. 5960-5967.

222 Cf. Lamba and Karimi (2002a) pp. 779-789 and Lamba and Karimi (2002b) pp. 790-800.

223 Cf. Giannelos and Georgiadis (2002a) pp. 2178-2184.

224 Cf. Giannelos and Georgiadis (2002b) pp. 2431-2439.

225 Cf. Wang and Guignard (2002) pp. 113-126.

A single-level MIP model formulation is proposed by Chen et al. (2002). Their model formulation relies on a new time representation. A number of time slots is defined which equals the number of orders during the time horizon. Two sets of bi-indexed binary variables are defined which relate orders (rsp. resources) and time slots. Sequencing constraints guarantee that a resource m can only be assigned to a time slot t if no resource $n>m$ is assigned to a time slot $s<t$. Consequently a time slot with a lower ordering number can lie well behind a time slot with a higher ordering number. Therefore, time slots inherit only their natural ordering for each resource independently. This modeling of time restricts the applicability of the proposed MIP model formulation to single-stage problems. Furthermore, batch processing times are assumed to be known a priori.[226]

The MIP model formulation by Grunow et al. (2003) uses two time scales. One of them is predetermined and associated with demand data, whereas the second one is continuous. The first time scale is used to match production with final demands. Consequently, inventory holding costs, which are calculated based on this time scale, are only included in the objective function for end products. The remaining part of the model formulation is based on the continuous time scale. The MIP model formulation is embedded into a three-stage approach consisting of aggregation, solution of the proposed MIP model formulation for several times and post-optimization to tackle larger problem sizes.[227]

Maravelias and Grossmann (2003a) present a MIP model formulation based on the STN representation. Their model formulation makes use of cont1 as well as of cont2. The reason for this is, that they want to take advantage of the smaller number of time points needed in cont2, but at the same time need to model zero-wait policies for perishable intermediates, which requires to take end points of certain tasks into account. Their model formulation allows for variable batch processing times and sequence dependent setup times. They compare their model with four other models based on a continuous time representation.[228] As only small examples are used to assess computational performance, all model formulations yield the same (optimal) solution, but the proposed formulation appears to be fastest. Especially the valid inequalities, which are basically constraints tightening the capacity allocation, seem to have a great impact.[229]

Table 4-3 summarizes the results of this literature review on continuous time model formulations.

An important advantage of continuous time model formulations is that variable batch processing times may be accounted for easily, thereby resulting in a more realistic description of the planning problem. Processing times are variable in all model formulations except the ones by Chen et al. (2002), Grunow et al. (2003), Hui and Gupta (2001) and Jordan (1996). But there are also several drawbacks to this representation of time.

---

226 Cf. Chen et al. (2002) pp. 1249-1260.

227 Cf. Grunow et al. (2003) pp. 109-141.

228 Their comparison comprises the model formulations of Schilling and Pantelides (1996), Ierapetritou and Floudas (1998a), Castro et al. (2001a) and Lee et al. (2001).

229 Cf. Maravelias and Grossmann (2003a) pp. 3056-3074.

Table 4-3: Classification of literature (continuous time model formulations).

Author(s)	Process Repre-sentation	Process characteris-tic	Model-ing of time	Solution method
Castro et al. (2001a)	RTN	batch	(cont2)	MIP
Castro et al. (2004)	RTN	batch, cont.	(cont2)	MIP
Chen et al. (2002)	-	batch	-	MIP, heuristic
Giannelos and Georgiadis (2002a)	STN	batch	event	MIP
Grunow et al. (2003)	-	batch	cont2	MIP, algorithm
Hui and Gupta (2001)	-	batch	-	MIP, heuristic
Ierapetritou and Floudas (1998a)	STN	batch	cont2	MIP, algorithm
Ierapetritou and Floudas (1998b)	STN	cont.	cont2	MIP, algorithm
Ierapetritou et al. (1999)	STN	cont.	cont2	MIP, algorithm
Jordan (1996)	-	batch	cont.	heuristics
Karimi and McDonald (1997)	-	-	event	MIP
Lamba and Karimi (2002a, 2002b)	-	-	cont1	MIP, heuristic
Lee et al. (2001)	STN	batch	cont1	(MINLP), MIP, algorithm
Majozi and Zhu (2001)	SSN	batch	-	MIP
Mockus and Reklaitis (1999a, 1999b)	STN	batch, cont.	cont1	MINLP
Maravelias and Grossmann (2003a)	STN	batch	cont1, cont2	MIP, cuts
Schilling and Pantelides (1996)	RTN	batch	cont1	MIP, algorithm
Wang and Guignard (2002)	-	batch	cont2	MIP
Zhang and Sargent (1996)	RTN	batch, cont.	-	MINLP, MIP

In all continuous time representations the number of time points needs to be determined beforehand: A too small number will not yield the optimal solution and a too big number will unnecessarily increase the model size and therefore affect the capability to solve the model formulation. Therefore, often an iterative procedure is proposed to determine the necessary number of time points.[230]

Although continuous time representations require far less time points than discrete time representations require periods, they often suffer from large integrality gaps due to poor LP relaxations.[231] This stems from the fact that start (and end) times of tasks need to be coupled with time points, which is done by so-called big-M constraints. Thereby, the advantage gained due to the decreased number of periods is often lost.

Moreover, to model interactions between tasks (multi-level case) or between resources (multi-resource case) a common time grid needs to be utilized, increasing the number of time periods / event points even further.

Finally, inventory holding costs cannot be accounted for in continuous time model formulations easily. Grunow et al. (2003) and Karimi and McDonald (1997), who both include this cost component into their objective function, rely on a second discretized time grid for their computation.[232] Mockus and Reklaitis

230 E.g., Ierapetritou and Floudas (1998a) p. 4349 and Castro et al (2001a) pp. 2066-2067.

231 Cf. Maravelias and Grossmann (2003a) p. 3057.

232 Cf. Grunow et al. (2003) p. 125 and Karimi and McDonald (1997) p. 2708.

(1999a) accept that their objective function gets non-linear.[233] All other model formulations dodge objective functions including this component and propose to maximize revenue or minimize tardiness instead.

Among the literature reviewed, minimal campaign quantities are only considered by Karimi and McDonald (1997) and the two papers of Lamba and Karimi (2002a, 2002b).[234]

233 Cf. Mockus and Reklaitis (1999a) p. 201.

234 Cf. Karimi and McDonald (1997) p. 2705, Lamba and Karimi (2002a) pp. 782-783 and Lamba and Karimi (2002b) p. 791.

5 Planning Framework and Solution Techniques

Before the proposed solution procedure is explained in detail, this chapter contains some thoughts on the planning situation and methodology as well as on applicable solution techniques. The concept of Advanced Planning Systems (APS) is introduced briefly as it provides an environment, for which the proposed solution procedure is well-suited.

Furthermore, the two fundamental building blocks the solution procedure is based upon are presented. These are mathematical programming and decomposition.

5.1 Planning Framework

Two planning philosophies, simultaneous planning and successive planning have been widely discussed in literature.[235] Whereas in simultaneous planning all decisions are taken simultaneously, the concept of successive planning aims at decomposing the decision situation into (independent) subproblems, solving these and constructing an over-all solution by combining the solutions of these subproblems. As most practically relevant decision situations are too complex to be tackled with a simultaneous planning approach, successive planning, which usually follows a (natural) hierarchical structure of the decision problem, is most often used.

In supply chain management, planning tasks can be structured along the two dimensions "planning horizon" and "supply chain process" to yield the so-called supply chain planning matrix (Fig. 5-1).[236] This matrix represents such a hierarchically organized planning system. In Fig. 5-1 typical planning tasks that occur in supply chains are shown. The planning tasks are structured into decisions based on a long-term, mid-term and short-term planning horizon reflecting the different temporal influences of the decisions. The second dimension is based on the typical flow of goods through a supply chain and describes the different processes (procurement, production, distribution and sales) involved.

The assignment of planning tasks to positions in Fig. 5-1 is not fixed though and may change in different decision situations. For example, the decision upon lot sizes is regarded as a short-term planning task of production in Fig. 5-1. In the process industries, which will be the primary area of application of the solution procedure devised here, lot-sizing is often regarded as a mid-term planning prob-

---

235 E.g., Stadtler (1988) pp. 21-29.

236 Cf. Rohde et al. (2000) pp. 10-15 and Fleischmann et al. (2002) pp. 76-82.

lem, because production is characterized by long production runs and setup operations that are far from negligible. To accommodate these important characteristics a longer planning horizon is deemed necessary.

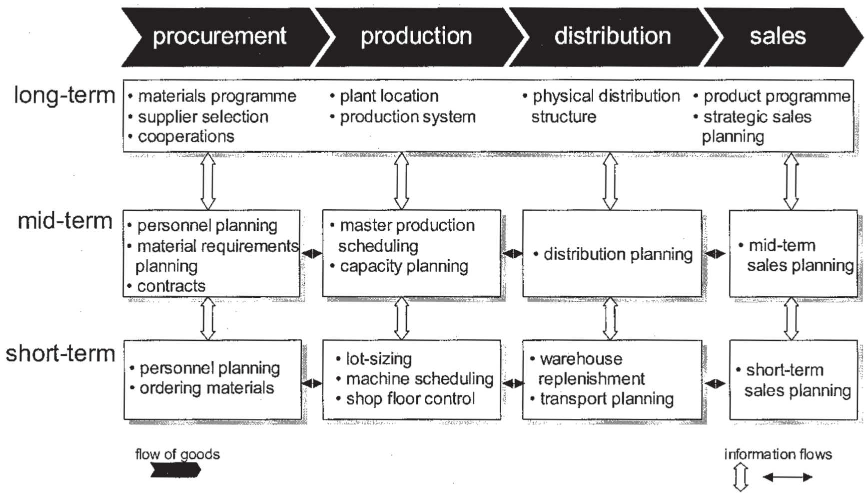


Fig. 5-1: Supply Chain Planning Matrix (planning tasks, Fleischmann et al. (2002) p. 77).

Advanced Planning Systems, which are software tools that support decision making in supply chains, have been developed by different software companies (e.g., AspenTech, i2 Technologies, Peoplesoft, SAP). In most of these systems the planning functionality has not been built into one powerful software application, but the functionality has been split up into several modules which cover the supply chain matrix. Fig. 5-2 provides generic, vendor independent names for these modules.[237]

The strategic network planning module covers the strategic long-term decision making level across all processes. With its help the structure of the supply network is determined (plant location, distribution system) as well as the product program. The demand planning module is used for short-term and mid-term sales planning. For short-term sales planning it is sometimes supported by a demand fulfillment and available-to-promise (ATP) module, which has the purpose to match inventories and production orders with demands, if customers require reliable quotes with only short notice. The master planning module coordinates procurement, production and distribution. This is usually done on a mid-term level. Here, the master production schedule is fixed. The short-term planning level needs to be anticipated at this planning level. Consequently, the most important characteristics of lower planning levels need to be modeled here. Distribution planning and transport planning cover distribution related planning tasks, the latter one on a more detailed

237 Cf. Rohde et al. (2000) pp. 10-15 and Meyr et al. (2002) pp. 99-101.

level (scheduling of transports, vehicle loading and routing). Production planning and scheduling on the other hand are the two modules to support production related issues. Finally, purchasing and material requirements planning support the short-term procurement of materials.[238]

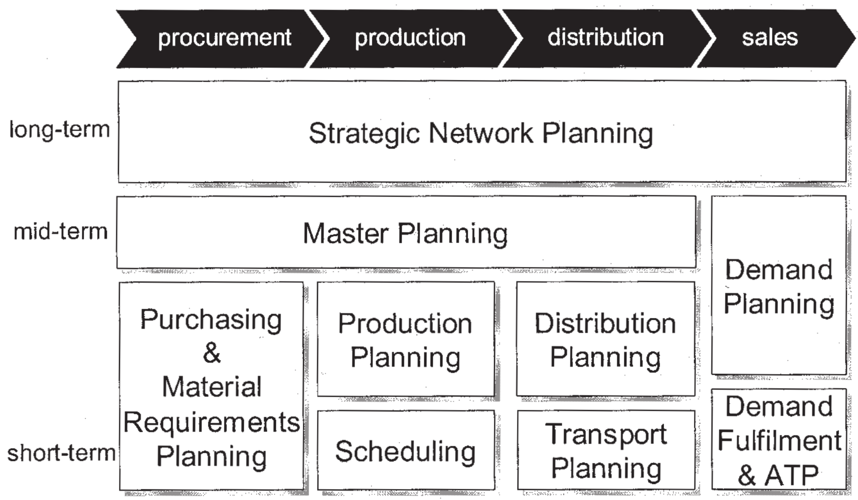


Fig. 5-2: Supply Chain Planning Matrix (APS modules, Meyr et al. (2002) p. 99).

The problem considered in this thesis is a production planning problem and – depending on the specific planning situation – can be attributed to the production planning or scheduling module or, if its decisions have a mid-term effect, to the master planning module. Its key components are products, resources and time. If their interaction can be anticipated fairly well by the mid-term planning level, it suffices to model their interaction explicitly only at the short-term planning level. On the other hand, if this is not possible, mid-term planning will result in plans that are either too ambitious and not feasible for the short-term planning level or – not even better – too conservative leading to an undesired under-utilization of resources. In this case, and especially if production decisions have a mid-term influence due to long production runs and substantial setup operations, planning has to be done on the master planning level.

In model based planning, an important cornerstone is that the real decision situation is represented adequately by the model, because otherwise the solution to the model cannot provide much benefit. As we have seen thus far with respect to the modeling of time, standard lot-sizing models do not provide an adequate

[238] For a detailed description of all modules see, for example, Stadtler and Kilger (2002). Neumann et al. (2002, pp. 253-258) present a slightly different framework with a focus on process industries.

model of the decision situation, if setup operations are substantial (with respect to the length of the periods) and thereby result in a representation defect.[239]

5.2 Solution Techniques

To overcome the aforementioned representation defect in various settings will be the focus of the proposed modeling and solution approach. A desirable attribute of the solution approach would be to incorporate additional restrictions easily as well as to allow a choice between the proposed modeling options, if not all of them are required in a specific situation.

Special purpose heuristics, which have been devised for many problem settings – although they may be customized to some extent – generally will only perform efficiently, if the underlying decision situation is matched by their premises. Thus, they are generally not the best choice here.

Mathematical programming on the other hand is a solution technique that fulfills this precondition. Additional restrictions are easily added to a model formulation as additional constraints. Thereby, mathematical programming based approaches are generally applicable.[240] Furthermore, the complete model formulation can be set up in a modular fashion such that several modules (sets of constraints) may be combined freely.[241] Additional advantages of mathematical programming based approaches are:[242]

- Optimal solutions are guaranteed to be found (if enough computational time can be spent).
- A feasible solution will be found, if one exists.
- Bounds on the objective function value are available (and improving) any time in the solution procedure (performance guarantee).
- Objectives may be altered easily, respectively the objective can be made up of differently weighted components.

Criticism on mathematical programming based approaches mainly considers scalability. Even as computational performance of hardware and software is constantly improving, model formulations with several hundred binary (or integer) variables constitute still a challenge.[243] Even though smaller practical problems might be solved with standard software by a good mathematical programming model formulation, decomposition will be treated as the second major building block of the devised solution procedure. Thus, also larger problems can be tackled by the proposed modeling and solution approach.

---

239 See chapters 2-4.

240 Cf. Clark and Clark (2000) p. 2289.

241 Such an approach is used in e.g. Wolsey (2002).

242 Cf. Zentner et al. (1994) p. 260, Pekny and Reklaitis (1998) p. 104 and Miller et al. (2000) p. 3.

243 E.g., Plapp (2003) p. 19.

5.2.1 Mathematical Programming

Mathematical programming has seen a lot of progress in recent years. Nevertheless, even in "spite of the remarkable improvements in the quality of general purpose mixed integer programming software, the effective solution of a variety of lot-sizing problems depends crucially on the development of tight formulations for the special problem features occurring in practice."[244] As there are several excellent text books on mathematical programming available,[245] this section will only focus on some selected topics, which will be exploited by the proposed modeling and solution approach, and not cover the whole area of mathematical programming. Especially, the focus will be on mixed integer programming model formulations which is a subset of mathematical programming in its entirety.

Finding a good model formulation often does not mean using the simplest formulation. Using different (or additional) sets of variables will often improve the model formulation in terms of solution quality and speed. This approach which is often referred to as reformulation or as extended formulation will be analyzed first. Secondly, it has proven advantageous to find model formulations which are as tight as possible, what means that the gap between the optimal solution of their linear relaxation and their optimal solution – which in a sense is an indicator of the size of the search space the branch-and-bound process has to cover – should be as small as possible. Last, additional techniques like preprocessing are discussed briefly followed by an assessment of the capabilities of state-of-the-art standard software solvers.

5.2.1.1 The "Art of Modeling" – Extended Formulations

There does not exist a clear methodology on how to obtain a good model formulation. Formulating a mathematical program is often an iterative process of (a) defining variables, (b) defining constraints using these variables so that the feasible points correspond to the feasible solutions of the problem and (c) defining an objective function using these variables.[246] However, for each specific problem several (an infinite number of) model formulations exist. This is illustrated in Fig. 5-3. All three polyhedra (P_1, P_2 and P_3) cover the same set of feasible integer solutions. Obviously, in this case P_3 is the best formulation (an ideal formulation), because its extreme points are integer solutions and thus the solution of the linear relaxation of formulation P_3 will yield an integer point. Although there exists such a formulation, i.e. the convex hull, for any problem, to obtain it may require too much effort to be an effective strategy.

A standard modeling technique is to extend the model formulation by replacing variables or adding new sets of variables. Several reformulations exist with respect

244 Belvaux and Wolsey (2001) p. 993. Overviews about these recent technological developments are given by Bixby et al. (2000) and Johnson et al. (2000).

245 E.g., Kallrath (2002a), Kallrath and Wilson (1997), Nemhauser and Wolsey (1988) and Wolsey (1998).

246 Cf. Wolsey (1998) p. 5.

to lot-sizing model formulations.[247] The most common approach is to use the so-called "simple plant location" (SPL) formulation.[248] This approach will be demonstrated on the most basic lot-sizing problem CLSP introduced in section 2.2 ((2-1)–(2-6)).

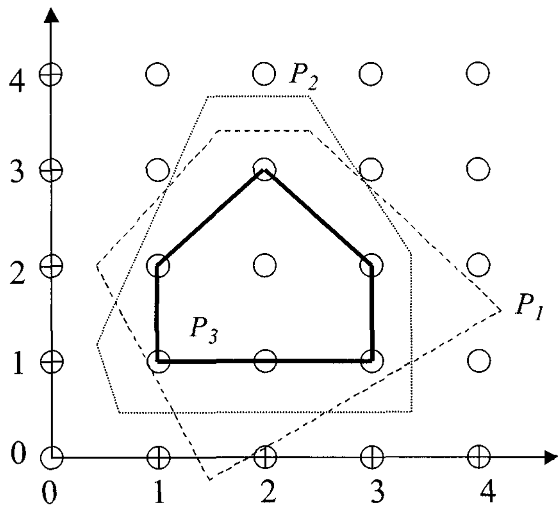


Fig. 5-3: Good and ideal model formulations (slightly adapted from Wolsey (1998) p. 15).

Model CLSP (SPL representation):

$$\text{Min} \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} \sum_{\substack{s\in\mathcal{T}\\ s>t}} h_{jt} \cdot (s-t) \cdot X_{jts} + \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} sc_j \cdot Y_{jt} \tag{5-1}$$

$$\sum_{j\in\mathcal{J}} \sum_{\substack{s\in\mathcal{T}\\ s\geq t}} a_j \cdot d_{js} \cdot X_{jts} + \sum_{j\in\mathcal{J}} st_j \cdot Y_{jt} \leq c_t \qquad \forall\, t\in\mathcal{T} \tag{5-2}$$

$$\sum_{\substack{t\in\mathcal{T}\\ t\leq s}} X_{jts} = 1 \qquad \forall j\in\mathcal{J},\, s\in\mathcal{T},\, d_{js}>0 \tag{5-3}$$

$$X_{jts} \leq Y_{jt} \qquad \forall j\in\mathcal{J},\, s,\, t\in\mathcal{T},\, t\leq s \tag{5-4}$$

---

[247] Cf. Eppen and Martin (1987) pp. 841-843, McKnew et al. (1991) pp. 287-288, Tempelmeier and Helber (1994) pp. 299-300, Stadtler (1996) pp. 562-572, Stadtler (1997) pp. 89-91 and Pochet (2001) pp. 73-77. A computational comparison has been carried out by Stadtler (1996, pp. 574-579) for the MLCLSP.

[248] Cf. Rosling (1986) p. 121, Pochet and Wolsey (1995) p. 253, Stadtler (1996) pp. 570-571 and Pochet (2001) pp. 73-74.

$$X_{jts} \geq 0 \qquad \forall j \in \mathcal{J},\ s,\ t \in \mathcal{T},\ t \leq s \qquad (5\text{-}5)$$

$$Y_{jt} \in \{0;1\} \qquad \forall j \in \mathcal{J},\ t \in \mathcal{T} \qquad (5\text{-}6)$$

Variables:

X_{jts} Fraction of demand of item j in period s which is produced in period t

The idea behind this model formulation is to replace variables X_{jt} for production quantities and I_{jt} for inventories by a new set of variables X_{jts} which is defined as the fraction of demand of product j in period s that is produced in period t ($t \leq s$). The information regarding inventory costs can now be derived from variables X_{jts} (first term in objective function (5-1)). Constraints (5-2) remain capacity constraints much similar to the ones of the original model formulation (2-3). The so-called big-M constraints that resulted in a rather weak linear relaxation of the original formulation (2-4) are now somewhat improved by a much lower value for "M" (5-4) (here: "1"). Finally, (5-3) are used to guarantee that all demands are fulfilled. (5-5) and (5-6) state the domain of the variables.

Fig. 5-4 shows a graphical representation of this approach for one product j. Production variables X_{jts} are represented by directed arcs from a production node (shown on the left side) to a demand node (shown on the right side). Each period, supposed it has positive demand, shows up twice in this representation (once for production and once for demand). Total production in a period X_{jt} is calculated by summing over all arcs leaving a production node. In the model formulation, constraints (5-4) take care, that a setup operation is carried out, if any leaving arc contains a positive production amount.

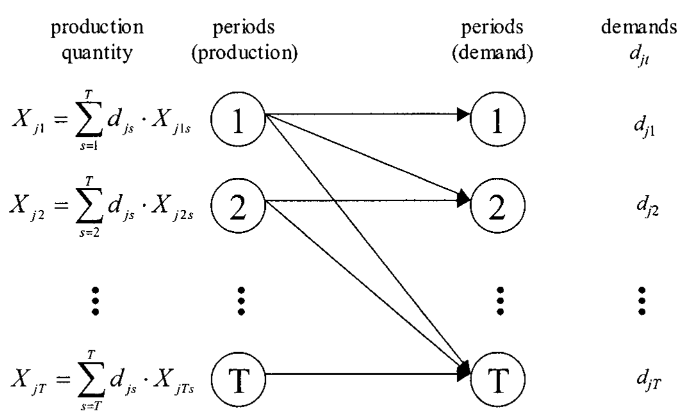


Fig. 5-4: CLSP in SPL representation.

This model formulation has advantages and disadvantages. Its main advantage is, that its linear relaxation provides a much tighter bound than the original formulation (which will be called I&L representation due to the variables used). On the downside, it requires much more variables and constraints to be formulated.

This basically constitutes the tradeoff to be analyzed when comparing MIP model formulations: A weaker model formulation with fewer variables and constraints might require a huge branch-and-bound tree to be explored until the optimal solution is found and proven to be optimal. On the other hand, a tighter model formulation using a greater number of variables and constraints might require a much smaller branch-and-bound tree, but the linear relaxation that needs to be solved in each node requires additional computational effort. Stadtler (1996) has shown this effect for different problem sizes for the multi-level CLSP.[249]

Regarding the SPL formulation it has been shown, that in the single-item uncapacitated case, the linear relaxation always leads to an optimal solution with all variables Y_{jt} being binary. So, this structure is an ideal formulation for some subproblems of the CLSP.[250]

Reformulation techniques have been applied to all kinds of lot-sizing problems.[251] Recently, Wolsey (2002) even claimed that "certain multi-item lot-sizing problems can now be solved just using standard reformulations and an off-the-shelf MIP solver."[252] This is due to the fact that during recent years a lot of knowledge has been accumulated on special structures of lot-sizing problems, which can be used like a toolkit to assemble good model formulations. Still, there remain many structures which still need in-depth exploration.[253]

Using extended model formulations or reformulations is one way of finding good model formulations. A different approach is to follow good modeling practice, which are ideas or guidelines that can help the solver avoiding numerical problems, reduce the computational burden at each node or help the solver to detect structures. A comprehensive collection of these modeling guidelines is given for example in the textbook by Kallrath (2002a)[254] and will therefore not be discussed in detail here.

5.2.1.2 Valid Inequalities

While reformulations change the variable space the model formulation is based on, applying valid inequalities to a problem is based on the initial variable space. The main idea of using valid inequalities lies in trying to find a description of the convex hull of a problem, while retaining the original variable space. The model formulation is simply extended by additional constraints (valid inequalities, cuts) here.[255]

---

249 Cf. Stadtler (1996) pp. 574-579.

250 Cf. Pochet (2001) p. 74.

251 Cf. Sox and Gao (1999) pp. 175-176 for a network representation of CLSPL.

252 Wolsey (2002) p. 1588.

253 Cf. Wolsey (2002) p. 1600.

254 Cf. Kallrath (2002a) pp. 102-149.

255 Research on valid inequalities was initiated by the seminal paper of Gomory (1958). Its first application in lot-sizing has been by Barany et al. (1984). Comprehensive in-

Valid inequalities are constraints, that do not exclude any feasible point, but cut off some part of the polyhedron that describes the linear relaxation of the problem. They are most helpful, if they are violated. A valid inequality is violated, if it is not fulfilled at the optimal solution of the relaxed problem and therefore, its addition would cut off this solution. Fig. 5-5 and (5-7)–(5-10) give an example.

$$\alpha \leq 10 \cdot \beta \tag{5-7}$$

$$0 \leq \alpha \leq 14 \tag{5-8}$$

$$0 \leq \beta \leq 4 \quad \text{and integer} \tag{5-9}$$

Constraints (5-7)–(5-9) describe the gray-shaded polyhedron in Fig. 5-5. Obviously, (5-10) is a valid inequality for this problem. It does not cut off any feasible point, but at the same time – if the optimal solution had been in the lower right corner of the polyhedron – would cut off the optimal solution of the linear relaxation.

$$\alpha \leq 6 + 4 \cdot \beta \tag{5-10}$$

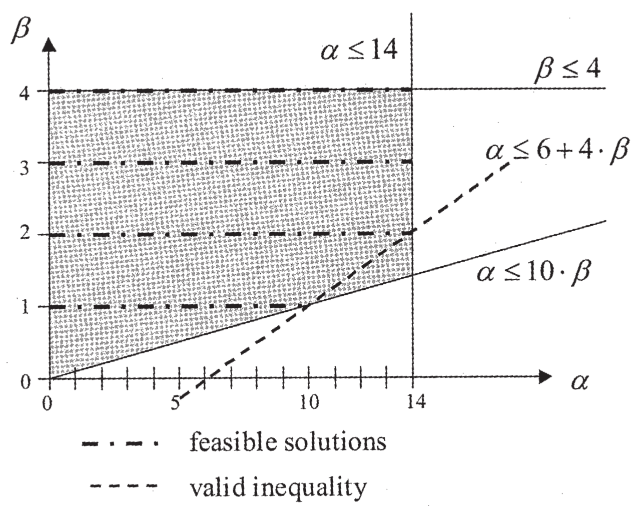


Fig. 5-5: Example for valid inequalities (slightly adapted from Wolsey (1998) p. 115).

Thereby, violated valid inequalities tighten the model formulation by improving the bound obtained by the solution of the linear relaxation of the problem. Consequently, research on valid inequalities has not only focused on deriving new classes of valid inequalities, but also on algorithms to decide whether any individ-

troductions into the theory of valid inequalities are given by e.g. Nemhauser and Wolsey (1988) pp. 205-291 and Wolsey (1998) pp. 113-161.

ual valid inequality is violated at a current node in the branch-and-bound tree. These algorithms are called separation algorithms, as they separate the violated valid inequalities from those that are not violated.[256]

Theoretically, one can use only the technique of valid inequalities without applying any branch-and-bound to obtain the optimal solution to a MIP problem. As long as any integer variable remains fractional, one would have to find a violated valid inequality, which would produce a different solution to the linear relaxation and iterate. On the other hand, the branch-and-bound method has proven to be quite successful in solving mixed integer programs. Therefore, a combination of both methods has been developed: branch-and-cut.

Branch-and-cut (B&C) as well as its special form cut-and-branch (C&B) combine the traditional branch-and-bound tree search with the use of valid inequalities. In B&C algorithms valid inequalities are added to the problem in any node of the search tree. These can either be inequalities that are only locally valid in branches diverging from the current node or globally valid inequalities that are also valid for the initial problem. In the first case, an efficient backtracking procedure needs to be implemented in addition. C&B is the special form of B&C where valid inequalities are only added to the model formulation in the root node of the search tree. This avoids any cut handling during the branch-and-bound process, but on the other hand incurs an additional computational burden, because many valid inequalities that are not violated (anytime) are carried through the whole solution procedure. Obviously, in C&B only globally valid cuts can be utilized.

Research on valid inequalities relevant to this thesis can be divided into two areas. The first area covers the derivation of new classes of valid inequalities in general. Valid inequalities belonging to this class are those, that can be applied to any mixed integer program.[257] The second area of valid inequalities are those that rely on special structures of the underlying problem. In this area, also a research stream emerged aiming at lot-sizing problems. Building on these, specialized branch-and-cut special purpose solvers have been developed.[258] Valid inequalities developed in this thesis are tailored to the MIP problems tackled here and will not be generalized. As valid inequalities have proven to be a viable method in improving performance of commercial MIP solvers, its providers keep adding functionality for generating valid inequalities and detecting those violated to their software. Hence, the search for classes of valid inequalities that might be suited for a special problem is no longer left to the user, but is done more or less automatically.

5.2.1.3 Further Enhancements

Whereas the first two approaches (reformulations and valid inequalities) in some way extend the initial problem formulation (by additional variables and/or constraints), also much effort has been made to reduce model sizes. These techniques are often subsumed under the title of preprocessing.

---

[256] Cf. Cordier et al. (1999) p. 337.

[257] Cf. Marchand et al. (2002) and Wolsey (2003) for an overview and recent developments.

[258] Cf. Belvaux and Wolsey (2000) 727-729.

Preprocessing[259] consists of methods to

- strengthen bounds on variables (fix variables),
- aggregate and/or substitute variables and constraints,
- eliminate redundant constraints,
- change coefficients of constraints and the objective function and
- detect infeasibilities.

Simple preprocessing techniques consider each constraint in isolation and check, e.g., whether it is consistent with the bounds of its variables. A more elaborate technique is probing. Here, variables are fixed to one of their bounds (0 or 1 for a binary variable) and all constraints of the formulation are checked simultaneously, whether this forces any other variables to their bounds.[260] In Bixby et al. (2000) an example is reported where probing rendered branch-and-bound obsolete.[261]

Preprocessing can be done on any MIP model formulation, but has also been applied to lot-sizing by exploiting knowledge about the model structure. Its first application has been by Maes et al. (1991).[262] We return to the model CLSP in SPL representation of section 5.2.1.1 ((5-1) –(5-6)) to illustrate their idea.

The model formulation builds on variables X_{jts} indicating the fraction of demand for product j in period s that is produced in period t. For example, if there is production in periods t and t+1 to cover demand of periods t+1 to t+3, the allocation of production quantities to production periods is arbitrary, allowing any combination. Now the idea is to drop variables from the formulation based on the fact that production from an "early" period to a "late" period may not be possible. The decision on each "early"/"late" period combination is based on the fact, that up to a certain "early" period – taking into account also demand fulfillment for the other products $k \neq j$ – only a limited quantity of j can have been produced (using the slack in capacity) resulting in inventory. Thus, any demands that occur in periods "later" than this maximal excess inventory can maximally cover cannot have been produced and their associated variables may be dropped from the model formulation.

Other enhancements that are often named as crucial improvements for the application of mathematical programming, but will not be explained in detail here, are the implementation of search strategies and primal heuristics.[263]

5.2.1.4 Capabilities of Standard Solvers

A lot of progress has been made in the last 15-20 years regarding the extent to which MIP model formulations can be solved now. This progress can be attributed

---

259 Cf. Savelsbergh (1994) pp. 445-452, Johnson et al. (2000) pp. 6-7 and Atamtürk and Savelsbergh (2003) p. 13.

260 Cf. Atamtürk and Savelsbergh (2003) p. 13.

261 Cf. Bixby et al. (2000) pp. 23-24.

262 Cf. Maes et al. (1991) pp. 138-139.

263 Cf. Bixby et al. (2000) pp. 15-16, Johnson et al. (2000) pp. 7-9 and Atamtürk and Savelsbergh (2003) pp. 5-13, 20-21.

to developments in algorithms, modeling and analysis of constraints as well as to the developments regarding hardware and (standard) software.[264]

For example, Table 5-1 shows the amount of time different versions of CPLEX needed to solve (the same) linear program. As these computations were performed on the same hardware, one has to take into account that during the same period the advance in computer hardware has been tremendous.

Table 5-1: Solution capabilities of CPLEX (LP, Bixby et al. 2000, p. 6).

Software	Year	Time (seconds)
CPLEX 1.0	1988	57840
CPLEX 3.0	1994	4555
CPLEX 5.0	1996	3835
CPLEX 6.5	1999	165

Today's standard solvers (CPLEX, LINDO, XPRESS-MP) offer much of the functionality introduced in the preceding sections right away: They incorporate functions to detect structure in the model formulation that allows for reformulations, they are able to generate valid inequalities and they perform different preprocessing techniques.[265]

The most difficult part for commercial solvers is obviously the first part, because to detect structure in a problem crucially depends on the model formulation the user has entered.[266]

Therefore, much more emphasis has been on incorporating the detection and use of valid inequalities, which have been an integral part of standard solvers for about five years.[267] Table 5-2 shows the impact the use of automatically generated valid inequalities has on six selected test instances. The different columns show the time (in seconds) and the number of nodes necessary to derive the optimal solution and prove optimality for each test instance. It is evident that in most cases computational effort is dramatically reduced by the incorporation of valid inequalities.

In addition to the generation of valid inequalities, standard solvers (like XPRESS-MP, which will be used for most of the computational testing in this thesis) offer additional functionalities. These concern the handling of valid inequalities or cut management. As argued above,[268] the generation of too many valid inequalities slows down the solution speed in any particular node, because of the size of the matrix. Therefore, not all valid inequalities that have been generated are put into the matrix, but most of them are put aside into a cut pool, from which valid inequalities can be drawn during the branch-and-bound search once they are violated.

264 Cf. Johnson et al. (2000) pp. 2-3.

265 Cf. Atamtürk and Savelsbergh (2003) for a survey of capabilities and advanced features of current commercial integer programming software systems.

266 Cf. Belvaux and Wolsey (2001) p. 996.

267 Cf. Wolsey (2003) p. 423.

268 See section 5.2.1.2.

Table 5-2: Impact of valid inequalities in CPLEX 7.5 (Atamtürk and Savelsbergh 2003, p. 20).

Problem	with cuts		without cuts	
	nodes	time (s)	nodes	time (s)
fixnet6	83	0.64	522	0.96
gesa2	148	1.90	101 852	264.14
gesa3	109	3.27	613	2.92
p0548	28	0.25	16 038	10.56
p2756	27	1.09	163 074	525.68
vpm2	8 812	7.69	222 698	104.60

Furthermore, if the user has derived a set of valid inequalities the solver cannot generate on its own, he can add these to the model formulation marked as "model cuts". The solver will then decide which of these model cuts are violated and put only those into the matrix, but store all the other ones in the cut pool. This leads to much smaller matrices compared to those in which the whole set of valid inequalities is put into the matrix instantly.[269]

Finally, Table 5-3 shows the impact preprocessing techniques have in a standard solver. The same test instances are used that have been utilized to demonstrate the effect of valid inequalities before. Although not as dramatic as the impact of valid inequalities, the integration of preprocessing techniques has still proven worthwhile.

Table 5-3: Impact of preprocessing in CPLEX 7.5 (Atamtürk and Savelsbergh 2003, p. 16).

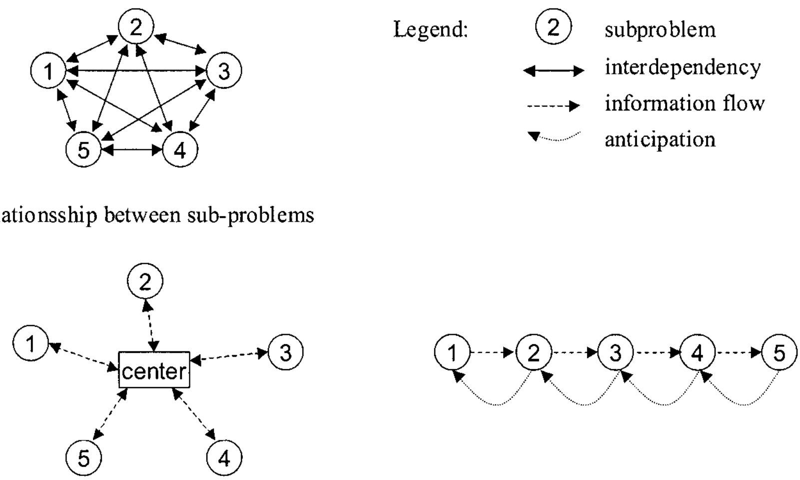
Problem	with preprocessing		without preprocessing	
	nodes	time (s)	nodes	time (s)
fixnet6	83	0.64	6 257	53.05
gesa2	148	1.90	52	0.92
gesa3	109	3.27	108	2.27
p0548	28	0.25	28	0.30
p2756	27	1.09	380	6.67
vpm2	8 812	7.69	18 492	22.57

5.2.2 Decomposition

Despite the remarkable advances in mathematical programming, a need for decomposition approaches remains when large industrial problems are tackled. In general, decomposition means (1) to divide a specific problem into a set of subproblems, (2) to solve these subproblems and (3) to build a solution to the initial problem by combining the solutions obtained for the subproblems. If these three steps are not designed with great care, solution quality will generally be bad – or worse, no feasible solution can be obtained in step (3).

269 Cf. Dash Optimization (2002) p. 168.

Fig. 5-6 shows two fundamental decomposition schemes, horizontal and vertical decomposition.[270] Usually, all subproblems are related somehow (Fig. 5-6 (a)). Therefore, it is important to pay attention to the interdependencies of the problem, when decomposing it into its subproblems. Interdependencies of the problems exist regarding the dimensions time, resources and products, which have been used before to classify lot-sizing problems in section 2.1.



(a) relationsship between sub-problems

(b) horizontal decomposition

(c) vertical decomposition

Fig. 5-6: Decomposition schemes.

Even more essential, these interdependencies are also important to observe when solving the subproblems. The reason for this is that a decision taken in one subproblem may influence decisions to be taken in another subproblem. Therefore, two types of coordination have to be distinguished. In horizontal decomposition (Fig. 5-6 (b)), all subproblems are equally important and no (natural) ordering of the subproblems exists. Here, a central unit takes care of the coordination of solutions of the subproblems. The information flows from the subproblems to the center are solutions, while the flow in the opposite direction is problem-defining data which might be influenced by solutions from other subproblems. Often, several iterations are necessary until all solutions of the subproblems are aligned.

In contrast, a (natural) ordering of the subproblems exists or can be determined in vertical decomposition (Fig. 5-6 (c)). Then, information regarding the solution of subproblems already solved is fed as data into the other subproblems. Thereby, coordination of solutions of the subproblems is achieved. On the other hand, decisions taken by a preceding subproblem may narrow the solution space for a subproblem, such that either only an inferior solution or no feasible solution can be

[270] Cf. Steven (1994) pp. 35-36.

found.[271] To avoid this, each subproblem has to anticipate the decisions of subsequent subproblems.

In lot-sizing literature as well as in literature dealing with related problems in the process industries, many decomposition schemes have been proposed. Most of them are based on time decomposition, but others are based on resource decomposition[272], product decomposition[273] or the relevance of decisions[274]. In the following, only those that are based on time decomposition combined with the application of mathematical programming techniques will be reviewed in more detail, because they offer the most promising results and fit best with the mathematical programming based modeling and solution approach proposed thereafter. Here, a (natural) ordering of subproblems often exists (by the ordering of the periods). Consequently, the proposed decomposition schemes mostly rely on some sort of vertical decomposition.

As linear programs of almost any practical size do not pose a too big challenge to today's mathematical programming software, the main goal in decomposing models is to reduce the number of binary and/or integer decision variables that have to be considered in each sub-model, because their number significantly influences the size of the branch-and-bound tree and therefore the running time of the algorithm.

Dillenberger et al. (1993) introduced such a decomposition scheme called "fix-and-relax", which was later also applied by Clark and Clark (2000).[275] Their decomposition approach for a CLSPL-like model considers the whole planning interval in each sub-model. In their first sub-model, integrality constraints for all periods except the first one are relaxed. In each successive sub-model, binary and integer variables are only present for one specific period t. Corresponding variables for periods $s<t$ have been fixed according to the solution of preceding sub-models, whereas integrality constraints remain relaxed for periods $s>t$. As sometimes infeasibilities may occur, they allow for backtracking. This means, the algorithm moves back one period at a time until a period is found, in which more than one feasible solution has been found. Starting from there, the next best solution is chosen and the algorithm continues.[276]

Two different approaches have been proposed by Blömer (1999).[277] In the first approach the number of binary variables is heuristically reduced. This reduction is based on a time-grid pattern and works as follows: Based on some insight or experience, it might appear that on a certain resource a setup operation occurs in approximately every third period. Then binary variables indicating setup operations on this resource are fixed to zero in two of three periods. The second approach is

---

271 Cf. Scholl (2001) p. 35.

272 Cf. Blömer and Günther (1998) pp. 252-253, Blömer (1999) pp. 110-112 and Wu and Ierapetritou (2003) pp. 1268-1270.

273 Cf. Blömer (1999) pp. 112-114 and Tempelmeier (2003) pp. 239-272.

274 Cf. Papageorgiou and Pantelides (1996) pp. 510-520.

275 Cf. Dillenberger et al. (1993) pp. 114-115 and Clark and Clark (2000) pp. 2302-2305.

276 Cf. Dillenberger at al (1994) pp. 282-283.

277 Cf. Blömer and Günther (1998) p. 253 and Blömer (1999) pp. 107-110, 114-117.

quite similar to the one by Dillenberger et al. (1993) although not periods, but sets of periods are considered in each sub-model. Furthermore, only those binary variables that have been "1" in solutions of a preceding sub-model are fixed, thereby allowing to schedule additional products in successive sub-models. As before, integrality constraints are relaxed in periods belonging to later sub-models.

This second approach bears much similarities to the "relax-and-fix" heuristic by Belvaux and Wolsey (2000), who also do not consider single periods like Dillenberger et al. (1993) but sets of periods (called subintervals) instead.[278] They also do not fix all variables based on the solution of preceding subintervals, but only those that have been "1" in the solution of a preceding subinterval. Thereby, they allow for additional production in periods preceding the current subinterval at the price of additional binary variables. Integrality constraints are relaxed in periods trailing the current subinterval.

Kelly (2002) somewhat broadens the methodology proposed before.[279] His idea is also to solve successive sub-models containing sets of periods (called time-chunks), with binary variables fixed to their respective solution of preceding time-chunks and integrality constraints relaxed in periods following the current time-chunk. But unlike Dillenberger et al. (1994)[280], who proposed to backtrack only if infeasibilities occur, he proposes a tree search with the nodes represented by feasible solutions of a time-chunk. The approaches described before followed a greedy heuristic: A sub-model was always based on the optimal or best known solution of the preceding sub-model. Although this procedure, which is called depth-first in Kelly's terminology, may often lead to a good solution, there might be the opportunity to find better solutions in successive sub-models, if not the best solution had been taken each time.

The approach by Federgruen et al. (2003) does not consider the whole planning interval in each sub-model like all the other approaches.[281] Instead, each sub-model spans over an increasing number of periods, from which only the last t periods contain binary or integer variables, whereas these variables in all periods preceding these last t periods are fixed from solutions of preceding sub-models.

Finally, a different approach is followed by the time decomposition heuristic of Stadtler (2003).[282] In his decomposition approach, sub-models (called lot-sizing windows) belong again to a set of periods, but consider the whole planning interval. Binary variables of periods preceding the lot-sizing window are fixed to their solution of preceding lot-sizing windows, whereas binary variables of periods later than the current lot-sizing windows are dropped from the sub-model. This can be done without loss of feasibility, as long as the binary variables are not bounded. This is the case here, if the MLCLSP without setup times is tackled. To overcome this feasibility issue when setup times are present, Stadtler (2003) proposes an anticipation of setup times arising in periods after the current lot-sizing window.

278 Cf. Belvaux and Wolsey (2000) p. 729.

279 Cf. Kelly (2002) pp. 2995-2999.

280 Cf. Dillenberger et al (1994) pp. 282-283.

281 Cf. Federgruen et al. (2003) pp. 3-4.

282 Cf. Stadtler (2003) pp. 490-494.

Especially in time decomposition schemes based on vertical decomposition (Fig. 5-6 (c)), anticipation is an important issue in the design of a decomposition procedure. The reason is that here a feedback loop is usually not envisioned, but decisions taken in one sub-model strongly affect the decisions of successive sub-models. In solution approaches for lot-sizing based on time decomposition, an effect calling for anticipation is well-known as truncated horizon or end effect.[283] It results, because in periods close to the planning horizon production and inventories go down, because a cost is usually associated with inventories and inventories at the end of the planning horizon are (in the view of the sub-model) useless. These effects can be dealt with by period overlap of sub-models,[284] assigning value instead of costs to ending inventories,[285] assigning bonuses to setup decisions and production close to the end of the planning interval[286] or by modifying the input data for successive sub-models.[287]

Interestingly, all approaches have in common, that each algorithm proceeds forward from the first period to the last. Sub-models are always based on single periods or sets of periods and except the approach by Federgruen et al. (2003) all remaining decomposition approaches incorporate some form of anticipation of successive sub-models, either by considering the whole planning interval in each sub-model, by explicit anticipation as described above or both. As all authors mentioned report great improvements regarding solution quality and solution speed, when applying their time decomposition approaches, time decomposition seems a promising option when designing the proposed modeling and solution approach.

283 Cf. Federgruen and Tzur (1994) p. 457, Stadtler (2000) pp. 318-320 and Fisher et al. (2001) pp. 679-680.

284 Cf. Blömer (1999) p. 115, Kelly (2002) pp. 2995-2996 and Stadtler (2003) pp. 490-491.

285 Cf. Fisher et al. (2001) pp. 679-684.

286 Cf. Stadtler (2000) pp. 320-321 and Stadtler (2003) pp. 491-492, 494.

287 Cf. Clark and Clark (2000) pp. 2294-2295.

6 Modeling and Solution Approach

As has been outlined in the last chapter, the solution approach consists of two parts. The major part are mathematical model formulations that can be used in standard MIP solvers like CPLEX or XpressMP to generate solutions. These are further enhanced by modeling tricks and valid inequalities to improve computational performance. Large industrial problems typically require some sort of decomposition to become tractable. Therefore, a decomposition heuristic which builds on these model formulations is proposed in the second part.

6.1 Model Formulations and Enhancements

The mathematical model formulations presented next are intended as building blocks. For ease of presentation this section is organized like chapter 3, in which the consideration of time continuity in time-indexed model formulations has been motivated by a thorough analysis of the representation defect usually encountered in this kind of model formulations.

First, attention is paid to the basic models that are used throughout this work, namely the PLSP and the CLSPL. Thereby, the modeling of setup states at period boundaries is covered. Then the focus is shifted to lot sizes / campaigns that last several periods. Following that, period overlapping setup times are examined and different resource utilization requirements are analyzed. Last, issues arising if several of these modeling options need to be combined are discussed.

In a problem setting with more than one resource, an additional index for resources needs to be introduced. The extension of the model formulations is straightforward taking into account that on each resource one setup state might be preserved rather independently. In section 7.5 an example will be presented.

6.1.1 Time Continuity – Setup States

The conservation of setup states across periods marks the first cornerstone to overcome the representation defect regarding time continuity in time-indexed model formulations. It is addressed here by defining a unique state for each resource at each period boundary. In small-bucket models setup state conservation is a constituent attribute, but also hybrid models allow for setup state conservation.[288]

[288] See also sections 2.3 and 2.4.

In this work it is assumed that setup states cannot get lost during idle time, but this assumption is not too restrictive. If idle time of a resource leads to a loss of the setup state, this can be handled by adding a dummy product which represents idle time and at the same time requiring a minimum utilization of resources.

Two basic models will be used throughout this thesis: The PLSP as the most universal small-bucket model suits perfectly as a basic model, because it offers sequencing and lot-sizing integrated into one model formulation. Setup states are preserved across period boundaries. The only disadvantage compared to other model formulations is the restriction of the PLSP that at most one setup operation can be performed in each period. This might require to define very short and therefore too many periods to obtain the "true" optimal solution to a specific production planning problem.[289] This solution is only guaranteed to be found by the PLSP, if the period length is shorter than the smallest lot size present in this solution.

At this point, the CLSPL offers remedy. By allowing setup operations for each product in each period and preserving setup states across period boundaries, in general less periods need to be defined compared to the PLSP to obtain a similar solution quality.[290]

Usually, the appropriate time discretization may not be freely chosen by the planner, but will at least in part be imposed by the circumstances and data of the planning problem. For example, period boundaries are defined by due dates of orders or any other relevant dates for the planner (e.g., closing dates at which high inventories need to be avoided). Starting from this natural time discretization, the PLSP can be used as a basic model, if the "true" optimal solution is expected to include only lot sizes that fit into this discretization. Otherwise a finer time resolution needs to be imposed and/or the CLSPL needs to be chosen as a basic model.

6.1.1.1 The Proportional Lot-Sizing and Scheduling Problem

As argued above, the PLSP can basically be used in its standard model formulation presented in section 2.3.3 ((2-20)–(2-27)). Apart from some standard valid inequalities only two minor modifications are proposed here, one for modeling purposes and the other one to enhance computational performance.

Modifications to Basic Model Formulation

For modeling purposes it is important, that at each period boundary a unique setup state is defined. Therefore, (2-24) are replaced by (6-1). With (2-24) setup states can be lost during idle time. On the other hand, (6-1) force the determination of a unique setup state. If resources lose their setup state during idle time this must be accounted for by a dummy product.

$$\sum_{j \in \mathcal{J}} Z_{jt} = 1 \qquad \forall\ t \in \mathcal{T} \qquad (6\text{-}1)$$

---

289 "True" optimal solution refers to the optimal solution based on a continuous time scale.

290 See also section 2.5.

In MIP model formulations it seems wise to choose the M in so-called big-M constraints as low as possible to obtain a tight formulation. In (2-23) M is determined as the maximal production quantity possible in a period based on available capacity and the production coefficient. Towards the end of the planning horizon, this estimate can be reduced, because the maximal production quantity is also restricted by the demand which needs to be fulfilled within the remaining periods up to the planning horizon.

$$X_{jt} \leq \min\left\{\frac{c_t}{a_j}, \sum_{\substack{s\in T \\ s\geq t}} d_{js}\right\} \cdot (Z_{jt} + Z_{jt-1}) \qquad \forall\ j\in\mathcal{J},\ t\in\mathcal{T} \qquad (6\text{-}2)$$

No more alterations to the PLSP are made. Other reformulations[291] will not be proposed, because in preliminary computational tests these did not show a superior performance.

Valid Inequalities

Two types of valid inequalities will be used throughout the computational tests in conjunction with the basic PLSP model formulation. For these and other valid inequalities the model cut option of XpressMP will be activated.[292]

Both types of valid inequalities have been known for long and have been shown to suffice to describe the convex hull of solutions of uncapacitated single item problems if the cost structure observes the Wagner-Whitin condition.[293]

The first one couples continuous inventory variables I_{jt} with the binary variables for setup operations and setup states. The rationale behind this constraint is, that a lower bound is imposed on the inventory variable if no production takes place within a certain interval of periods (t+1..t+p). Production of a certain product is not possible if no resource is set up for this product at the beginning of the interval (Z_{jt}=0) and no setup operation for this product is performed within the time interval.

$$I_{jt} \geq \sum_{s=t+1}^{t+p} d_{js} \cdot (1 - Z_{jt} - \sum_{r=t+1}^{s} Y_{jr}) \qquad \forall\ j\in\mathcal{J},\ t=0..T-1,\ p=1..T-t \qquad (6\text{-}3)$$

The second set of valid inequalities represents a simple relation between variables for setup operations Y_{jt} and setup states Z_{jt}. A setup operation in a certain period is always followed by a corresponding setup state at the next period boundary, because only this setup state allows for production (6-2). The validity is obvious: Otherwise, the setup operation has been wasted.

---

291 E.g., Belvaux and Wolsey (2001) p. 999.

292 See also section 5.2.1.4.

293 Cf. Pochet and Wolsey (1994) pp. 301-303. The Wagner-Whitin condition imposes a relation between cost coefficients such that the sum of production costs in period t and holding costs in t are more expensive than production costs in period t+1. Particularly, the Wagner-Whitin condition is always fulfilled if production costs are period independent and holding costs are positive.

$$Y_{jt} \leq Z_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \qquad (6\text{-}4)$$

6.1.1.2 The Capacitated Lot-Sizing Problem with Linked Lot Sizes

The basic model formulation of the CLSPL presented in section 2.4.1 allows for several improvements. First, a tighter model formulation is derived by introducing a new type of variables. This extended model formulation is further improved by the addition of three types of valid inequalities. These measures have proven to be effective, if the CLSPL needs to be solved by a standard MIP solver.[294] Should also other elements like period overlapping lot sizes or setup times be required, this modeling approach lacks one important attribute, the unique setting of setup states. How to deal with that will be the last topic of this subsection.

Extended Model Formulation

The extended model formulation is based on a new set of variables VV_{jt}. These variables replace variables V_t of the original formulation ((2-28)–(2-34)). Whereas variables V_t (=1) indicate, that only one product is produced in period t and that a setup state is carried from period t–1 to period t+1, variables VV_{jt} (=1) indicate the same with reference to a specific product. They are therefore product dependent versions of the former. Although they take only values "1" and "0" in feasible solutions, they do not need to be defined binary explicitly.

With this definition in mind, constraints (2-29)–(2-31) and (2-33) can be replaced by (6-5)–(6-8).

$$Y_{jt} + W_{jt} + \sum_{\substack{k \in \mathcal{J} \\ k \neq j}} VV_{kt} \leq 1 \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \qquad (6\text{-}5)$$

$$W_{jt} \leq Y_{jt-1} + VV_{jt-1} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \setminus \{1\} \qquad (6\text{-}6)$$

$$VV_{jt} \leq W_{js} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \setminus \{1, T\}, s = t..t+1 \qquad (6\text{-}7)$$

$$VV_{jt} \geq 0 \qquad (VV_{j1} = 0, \;\; VV_{jT} = 0) \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \qquad (6\text{-}8)$$

VV_{jt} Single-product indicator, which indicates that (a) only product j is produced in period t, and (b) its setup state is brought from period t–1 and carried into period t+1 (VV_{jt}=1); otherwise (VV_{jt}=0)

Constraints (6-5) correspond to (2-31). They are derived by replacing V_t in the original formulation by VV_{jt} and by subtracting W_{jt}–VV_{jt} (≥0, due to (6-7)) on the RHS, which strengthens the constraints. They are valid, because in a certain period t there can be either a setup operation for product j, a link for product j into period t, single-item production of any product k≠j or none of these options, but never two of them. This is not obvious, because a situation may arise, in which a setup state is carried into a period, some other products are produced and at the end of the period, the first product is set up again, because its setup state is re-

294 Cf. Suerie and Stadtler (2003) pp. 1048-1052.

quired again in the subsequent period. This issue is resolved in the CLSPL such that the incoming setup state is lost due to (2-28). This does not affect the optimal solution of the CLSPL, because it does not make any difference if production is shifted from the beginning of a period towards the end.[295] On the other hand, this remains an issue, if e.g. certain campaign quantities need to be matched.

Constraints (6-6) correspond to (2-29) of the original formulation. Their validity is obvious, as a setup carry-over from period t to $t+1$ is only possible if the corresponding product has been set up in period t or the setup state has already been carried over from $t-1$ to t, implying single-item production of the product in question in t.

Finally, constraints (6-7) relate link variables W_{jt} and single-item production variables VV_{jt}. Single-item production forces both link variables (into and out of the period) to "1". (6-8) state the domain of variables VV_{jt}.

An additional set of constraints corresponding to (2-30) is not necessary, because it will be dominated by (6-5) and (6-6).[296] Summing up, the extended model formulation requires $(J-1)\cdot T$ more continuous variables and $J\cdot T$ additional constraints, but is much tighter than the original formulation (see Table 7-11).

Valid Inequalities

Three types of valid inequalities can be added to the model formulation. The first type (preprocessing-inequalities) rests on the following idea: If capacity is tight, it might be ruled out that there is production of only one product in a certain period. This information, which is solely derived from the problem data, is used to limit the domain of single-product indicator variables VV_{jt}. Of course, this type of inequality is effective only in tight capacity situations, but these are usually the more difficult problems to solve.

Before the valid inequalities are presented, the underlying concept will be shown by means of an example. Consider three products which have to be produced on a shared resource. Production coefficients for these products as well as demands (with initial inventory already absorbed in the demand figures) are provided in Table 6-1. Taking into account available capacity and that demand has to be fulfilled (no backlogging), one can compute cumulative slack capacity.[297] Cumulative slack capacity in period t is defined as the amount of capacity that has not been used up to period t if there has been lot-for-lot production up to period t and setup times have been neglected. For example, in t=2 cumulative slack capacity amounts to 50 (=2·100–20–30–20–20–40–20).

In the example, several conclusions can be drawn: First, in period t=2 production of only one product is not possible. The reason is, that to produce only one product in t=2 a minimum of 40 units must have been produced beforehand, but

[295] Such a case has been constructed by Suerie and Stadtler (2003) p. 1053.

[296] This is shown easily. First, V_t in (2-30) is replaced by VV_{jt}. Next, VV_{jt} is replaced by $W_{jt+1}-Y_{jt}$ (6-6). Then W_{jt+1} is subtracted on both sides. The remaining inequality is valid because of (6-5). Q.e.d.

[297] The concept of cumulative slack capacity is also used by Maes et al. (1991) to derive valid inequalities for the serial MLCLSP. Cf. Maes et al. (1991) pp. 138-139.

cumulative slack capacity in t=1 is only 30. Second, in period t=3 it is impossible to produce only j=1 or j=3. The reasoning here is similar. To produce either j=1 or j=3 60 units must have been produced in earlier periods. Again this is not allowed as slack capacity in t=2 is only 50 units. Summing up, by calculating slack capacity it is possible to rule out single-item production for distinct periods (here: t=2) or for certain products in distinct periods (here: j=1 and j=3 in period t=3). Thus, it is possible to obtain upper bounds (here: "0") for certain variables VV_{jt}.

Table 6-1: Example (preprocessing-inequalities).

	a_j	$d_{j,1}$	$d_{j,2}$	$d_{j,3}$
Product j=1	1	20	20	20
Product j=2	1	30	40	40
Product j=3	1	20	20	20
Available capacity		100	100	100
Cumulative slack capacity		30	50	

Generally speaking, the argument is as follows (U^1 [U^2] denotes the length of the interval under consideration): If cumulative slack capacity (up to period t–1) is less than the amount that has to be pre-produced to allow for single-item production of just one product in the interval under consideration [t; $t+U^1-1$], then at least two products have to be produced in the interval [t; $t+U^1-1$]. This implies that at least one setup activity has to be performed, which implies that not all periods of the interval [t; $t+U^1-1$] can have single-item production. Thus, the valid inequalities can be stated as follows.

$$\sum_{s=t}^{t+U^1-1} VV_{js} \leq U^1 - 1 \qquad \forall j \in \mathcal{J}, =2..T\text{-}U^1+1,\ U^1=1..3, \text{ if} \qquad (6\text{-}9)$$

$$\sum_{s=1}^{t-1} c_s - \sum_{s=1}^{t-1} \sum_{k \in \mathcal{J}} a_k \cdot d_{ks} - \sum_{\substack{k \in \mathcal{J} \\ \sum_{s=1}^{t-1} d_{ks} > 0}} st_k - \sum_{s=t}^{t+U^1-1} \sum_{\substack{k \in \mathcal{J} \\ k \neq j}} a_k \cdot d_{ks} < 0$$

$$\sum_{s=t}^{t+U^2-1} \sum_{j \in \mathcal{J}} VV_{js} \leq U^2 - 1 \qquad \forall\ t=2..T-U^2+1,\ U^2=1..3, \text{ if} \qquad (6\text{-}10)$$

$$\sum_{s=1}^{t-1} c_s - \sum_{s=1}^{t-1} \sum_{j \in \mathcal{J}} a_j \cdot d_{js} - \sum_{\substack{j \in \mathcal{J} \\ \sum_{s=1}^{t-1} d_{js} > 0}} st_j - \sum_{s=t}^{t+U^2-1} \sum_{j \in \mathcal{J}} a_j \cdot d_{js} + \max_{j \in \mathcal{J}} \left(\sum_{s=t}^{t+U^2-1} a_j \cdot d_{js} \right) < 0$$

With U^1 chosen as "1" in (6-9), single-product indicator variables VV_{jt} are forced to zero if the associated condition applies. This condition calculates cumulative slack capacity by summing up available capacity up to period t–1 (term 1) and subtracting those amounts of capacity that would have to be pre-produced to enable single-product production of item j. Demands up to period t–1 (term 2), minimum required setup times (term 3)[298] and those demands of period t that have to be pre-produced (term 4) are subtracted. Different values for U^1 extend this concept to consecutive periods of single-item production. Inequalities (6-10) are product independent variants of (6-9). As the effect of both of these constraints vanishes with increasing values for U^1 (rsp. U^2), these constraints are formulated for $U^1 \leq 3$ (rsp. $U^2 \leq 3$) only.

The second set of valid inequalities (inventory/setup-inequalities) transfers the set of valid inequalities presented for the PLSP (6-3) to the CLSPL. They are almost similar, only the naming of variables is different. Whereas Z_{jt-1} denotes if a setup state for product j prevails at the end of period $t-1$ and thus at the beginning of t, here W_{jt} denotes if the setup state of product j is linked from $t-1$ to t.

$$I_{jt} \geq \sum_{s=t+1}^{t+p} d_{js} \cdot (1 - W_{jt+1} - \sum_{r=t+1}^{s} Y_{jr}) \qquad \forall j \in \mathcal{J}, t=0..T-1, p=1..T-t \quad (6\text{-}11)$$

The last set of valid inequalities (capacity/single-product indicator-inequalities) establishes a relation between capacity constraints (2-3) and single-product indicator variables VV_{jt}. A version which does not require the definition of additional variables is (6-12).

$$\sum_{j \in \mathcal{J} \setminus \mathcal{JS}} (a_j \cdot X_{jt} + st_j \cdot Y_{jt}) \leq c_t \cdot \left(1 - \sum_{k \in \mathcal{JS}} VV_{kt}\right) \qquad \forall \mathcal{JS} \subset \mathcal{J}, t \in \mathcal{T} \setminus \{1,T\} \quad (6\text{-}12)$$

These valid inequalities state, that if there is single-product production of any item $k \in \mathcal{JS}$ in period t no capacity is available in this period for any product not contained in this subset of products ($j \in \mathcal{J} \setminus \mathcal{JS}$). These inequalities are clearly valid, as they are reduced to a weaker form of the basic capacity constraints (2-3) if no single-product indicator variable VV_{kt} takes a positive value.

Valid inequalities (6-12) may be strengthened by the introduction of additional variables XV_{jt}, defined as the production amount of product j in period t if this period is a single-product production period.

$$XV_{jt} \leq X_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \setminus \{1,T\} \quad (6\text{-}13)$$

$$XV_{jt} \leq \min(\frac{c_t}{a_j}, \sum_{s=t}^{T} d_{js}) \cdot VV_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \setminus \{1,T\} \quad (6\text{-}14)$$

$$\sum_{j \in \mathcal{J}} (a_j \cdot X_{jt} + st_j \cdot Y_{jt}) \leq c_t \cdot (1 - \sum_{j \in \mathcal{J}} VV_{jt}) + \sum_{j \in \mathcal{J}} a_j \cdot XV_{jt} \qquad \forall t \in \mathcal{T} \setminus \{1,T\} \quad (6\text{-}15)$$

XV_{jt} Production amount of product j in period t, if period t is a period with single-product production

Constraints (6-13) and (6-14) define variables XV_{jt} and (6-15) give the valid inequalities. The reasoning here is to simply bound the LHS of capacity constraints (2-3) to the amount of single-product production if the period is a period in

298 Constraints (6-9) and (6-10) assume that no product is set up at the beginning of the planning interval. Otherwise the third term would need to be adapted accordingly.

which single-product production takes place. Again, if this is not the case, (6-15) reduces to a weaker form of basic capacity constraints (2-3).

Unique Setup States

As has been argued above, the tight model formulation presented lacks one attribute that becomes important, if e.g. campaign quantities have to be considered: The proposed model formulation does not guarantee that the setup state at each period boundary is defined properly. An example for this special case is presented by Suerie and Stadtler (2003, p. 1053). This does not affect the optimal solution of the CLSPL, but may affect the optimal solution if additional restrictions are applied.

To overcome this shortcoming, the basic CLSPL formulation requires several modifications. Constraints (2-28), (2-30) and (2-31) need to be replaced by (6-16) and (6-17).[299]

$$\sum_{j \in \mathcal{J}} W_{jt} = 1 \qquad \forall\, t \in \mathcal{T} \setminus \{1\} \quad (6\text{-}16)$$

$$W_{jt-1} + W_{jt} - Y_{jt-1} + Y_{kt-1} \leq 2 \qquad \forall\, j,k \in \mathcal{J}, j \neq k,\, t \in \mathcal{T} \setminus \{1\} \quad (6\text{-}17)$$

Constraints (6-16) guarantee that there is a unique setup state defined at each period boundary. Constraints (6-17), which replace (2-30) and (2-31) of the original model formulation, allow for the correct setting of the binary variables. Essentially they state that if both link variables (into and out of period $t-1$) are "1" for one product j, there has to be single-product production (every $Y_{kt-1}=0$) or there has to be a setup operation for product j in period t.

This model formulation cannot make use of single-product indicator variables VV_{jt}, which have been used to tighten the initial model formulation. The reason for this are constraints (6-5), which forbid that a setup state for a certain product j is carried into period t and also a setup operation for j is performed in t. In the extended model formulation, this setup carry-over is lost.[300] This has no consequence for the basic CLSPL, because it does not matter where production takes place within a period. On the other hand, this information is essential, if further extensions regarding time continuity are introduced. In consequence, this model formulation is less tight, because only one of the proposed valid inequalities does not make use of variables VV_{jt} and thus can also be used here: inventory/setup inequalities (6-11).

6.1.2 Time Continuity – Lot Sizes

In this section the focus will be on lot sizes which span over several periods. These will be reviewed in two steps. First, the PLSP will serve as the underlying

---

299 This model formulation is used by Sox and Gao (1999) p. 174 and Porkka et al. (2003) pp. 1137-1138.

300 Cf. the example by Suerie and Stadtler (2003, p. 1053).

basic model.[301] Second, modifications which are necessary if the CLSPL is chosen as the basic model will be discussed.

6.1.2.1 Basic Model: PLSP

With the PLSP as the basic model,[302] the integration of period overlapping lot sizes is straightforward, since the basic PLSP performs lot-sizing and sequencing simultaneously. The difficulty lies in (a) adding up the quantities produced in different periods which belong to one lot and (b) the extension that two consecutive lots of the same product may be a desirable outcome. The latter needs to be modeled explicitly, because more than one lot of a certain product might be produced directly after another, if its production quantity would otherwise exceed its maximum lot size. This will never occur in solutions to the basic PLSP, as it would incur a setup cost without any change in the setup state.

We will start with the introduction of the basic campaign restrictions followed by the consideration of the special case called "batch flow" (see section 3.3). Afterwards, some valid inequalities will be discussed and last, the extension that allows for consecutive lots of the same product will be presented.

Basic Campaign Restrictions

To compute lot sizes (or campaign quantities) new variables need to be introduced. These variables K_{jt} will be called campaign variables to distinguish them from production variables X_{jt}. The campaign variables K_{jt} will be defined to hold the cumulated lot size (or campaign quantity) of product j in period t of the last (or current) campaign. As long as j is produced current production of j is added to variable K_{jt}, whereas if production has ceased, variables K_{jt} will remain constant until they are reset to zero at the beginning of the next campaign of product j. Constraints (6-18) to (6-21) define the new set of variables K_{jt}.

$$K_{jt} \leq K_{jt-1} + X_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \qquad (6\text{-}18)$$

$$K_{jt} \geq K_{jt-1} + X_{jt} - maxlot_j \cdot Y_{jt+1} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \setminus \{T\} \qquad (6\text{-}19)$$

$$K_{jt} \leq maxlot_j \cdot (1 - Y_{jt+1}) \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \setminus \{T\} \qquad (6\text{-}20)$$

$$K_{j0} \leq minlot_j \cdot (1 - Y_{j1}) \qquad \forall j \in \mathcal{J} \qquad (6\text{-}21)$$

Data:

$minlot_j$	Minimal lot size (campaign quantity) for product j
$maxlot_j$	Maximal lot size (campaign quantity) for product j

---

[301] Some of these model formulations have been discussed (in less detail) in Suerie (2004) pp. 9-14.

[302] The basic PLSP is defined by objective function (2-20) and constraints (2-21), (2-22), (6-1), (6-2) and (2-25)–(2-27). Furthermore, the inequalities (6-3) and (6-4) are valid.

Variables:

K_{jt} Campaign variable for product j in period t (current campaign quantity up to period t)

Unless a new campaign of product j starts in the next period (Y_{jt+1}=1) current production (X_{jt}) is added to the campaign variables K_{jt} ((6-18) and (6-19)). These two constraints provide upper and lower bounds on the campaign variables K_{jt}. In this case, (6-20) and (6-21) take no effect. On the other hand, if there is a start of a new campaign, (6-19) is lifted (due to the last term on the RHS) and (6-20) which dominates (6-18) in this case forces the campaign variables to zero.

At the beginning of the planning horizon variables K_{jt} need to be initialized. This can be done by constraints (6-21).

With variables K_{jt} properly defined additional constraints can be identified to cope with the different restrictions posed to the decision problem by the production environment. As outlined in section 3.3, minimal lot sizes, maximal lot sizes and lot sizes which are multiples of an integer batch size are relevant restrictions to be considered here.

$$K_{jt-1} + X_{jt} \geq minlot_j \cdot \sum_{\substack{k\in\mathcal{J}\\k\neq j}} Y_{kt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \quad (6\text{-}22)$$

$$K_{jt-1} + X_{jt} \leq maxlot_j \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T}\setminus\{1\} \quad (6\text{-}23)$$

$$K_{jt-1} + X_{jt} = bs_j \cdot R_{jt} + S_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T}\setminus\{1\} \quad (6\text{-}24)$$

$$S_{jt} \leq bs_j \cdot \left(1 - \sum_{\substack{k\in\mathcal{J}\\k\neq j}} Y_{kt}\right) \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T}\setminus\{1\} \quad (6\text{-}25)$$

$$S_{jt} \geq 0 \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T}\setminus\{T\} \quad (6\text{-}26)$$

$$R_{jt} \geq 0 \quad \text{and integer} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T}\setminus\{T\} \quad (6\text{-}27)$$

Data:

bs_j Batch size for lots (campaigns) of product j

Variables:

R_{jt} Integer number of full batches produced in the current campaign of product j up to period t

S_{jt} Slack variable, residual quantity of the last batch of product j in period t which is not finished in t

To obey minimal lot sizes, (6-22) are necessary. They state that as soon as any new campaign of product k starts, the minimal lot size of product j has been produced. These constraints also hold true if product j is not produced right before k

because the re-initialization of campaign variables is just prior to its own next production start (6-20).

Maximal lot sizes are obeyed due to constraints (6-23). Any campaign may not exceed its maximal production quantity $maxlot_j$.

To deal with batch size restrictions is slightly more elaborate. Two new variables need to be defined, one of them being integer. In each period t the current campaign quantity (K_{jt-1}+X_{jt}) is split into two variables. One of them (R_{jt}) counts the number of full batches already produced in the current campaign and the second one (S_{jt}) takes the rest. This is done by constraints (6-24) and (6-25). The latter one takes care that neither more than a full batch is contained in slack variables S_{jt} nor any rest remains if production of another lot starts. Finally, (6-26) and (6-27) state the domain of variables R_{jt} and S_{jt}.

Following this line of arguments it is easily seen that obeying minimal lot sizes, maximal lot sizes and batch size restrictions can be achieved rather independently. If one of these restrictions is not present in the underlying decision problem the corresponding constraints ((6-22), (6-23), rsp. (6-24)–(6-27)) may be dropped from the model formulation.

Unfortunately, we are not finished yet. With the additional constraints ((6-18)–(6-27)) in place, the basic PLSP loses one of its properties, namely that two consecutive setup operations of the same product will not occur. This issue never materializes in the basic PLSP, because it would incur additional setup costs and therefore deteriorate the objective function value without any benefit (the setup state remains unchanged). Here, this additional setup operation can be utilized to reset campaign variables K_{jt} to avoid the correct verification of, e.g., the batch size restrictions. To avoid this case, (6-28) is added to the model formulation. (6-28) prohibits a new setup operation for product j, if the resource is already set up for product j.

$$Y_{jt} \leq 1 - Z_{jt-1} \qquad \forall\, j \in \mathcal{J},\, t \in \mathcal{T} \quad (6\text{-}28)$$

This modeling approach is fine as long as the aforementioned property of the basic PLSP is desirable. In many practical settings two consecutive setup operations of the same product will never be necessary. But this changes, if the maximal lot size of a product becomes a hard constraint, e.g., due to regulatory cleaning requirements for the equipment. Before this special case and its consequences will be analyzed in detail, the general model formulation is finished first.

As long as constraints (2-25), (6-1), (6-4) and (6-28) are present in the model formulation, variables Y_{jt} do not need to be defined as binary variables. Only if constraints (6-28) are skipped or constraints (6-4) are defined as model cuts, variables Y_{jt} need to be defined as binary variables.

In the above model formulation assumptions have been made with respect to the beginning and the end of the planning interval. These are easily adapted taking the actual decision situation into account.

Regarding the beginning of the planning interval it has been assumed that no machine state prevails and that therefore the resulting production plan will start with a setup operation. In practical settings the planner will experience that there is some frozen horizon based on production decisions taken in the last planning cycle which he is not allowed to alter at this point. Consequently, his planning interval will start at some point in time with a fixed machine state and probably some production already started. This is easily taken into consideration by providing initial setup states (Z_{j0}) and by setting appropriate values for variables K_{j0}. If a campaign looms into the planning interval, K_{j0} may be set to the amount produced prior to the planning interval and (if applicable) lower bounds on the production amounts X_{jt} of the first period(s) can be derived from shop floor decisions already taken (procured materials).

Regarding the end of the planning interval, campaign restrictions are not applied to the last campaign started in the planning interval (due to constraints (6-22) and (6-25)). The rationale here is that the campaign started in (one of) the last period(s) will probably fulfill the campaign restrictions after the planning horizon, i.e. in periods which are not considered yet. This is one way to overcome the so-called planning horizon or end-effect,[303] which is regularly observed in lot-sizing models.

Campaigns with Batch Flow Constraints

In order to include the batch flow scenario small changes to the inventory balance constraints (2-21) of the basic PLSP are necessary. They are altered to allow only completed batches to satisfy demand or build up inventories. This is done by adding the production amount which is kept in the slack variables S_{jt} one period later to inventory (6-29). In the period following the completion of a lot S_{jt} is set to zero due to (6-25) and thus the complete lot has entered the modified inventory balance constraints (6-29).

Constraints (6-30) are for initialization purposes.

$$I_{jt-1} + X_{jt} + S_{jt-1} - S_{jt} = I_{jt} + D_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (6\text{-}29)$$

$$S_{j0} = 0 \qquad \forall j \in \mathcal{J} \quad (6\text{-}30)$$

A shortcoming of this modeling approach is, that inventory can be held in the slack variables S_{jt}. This is possible only for the duration of a campaign, because as soon as another campaign starts, variables S_{jt} are reset to zero. The maximal extent of inventory held in the slack variables amounts to one batch due to (6-25). This situation will therefore have considerable effects only if the capacity situation is not very tight. Anyway, if this is an issue, it suffices to add the inventory balance constraints of the basic PLSP (2-21) and to replace inventory variables I_{jt} in (6-29) by a second set of inventory variables. Then, inventory considered in the objective function is calculated based on (continuous) production, but demand fulfillment is based on (batch) production.

Valid Inequalities

A first group of valid inequalities stems from the reasoning that whenever batch size restrictions are present, this leads to some inevitable base stock. This base

[303] Cf. Stadtler (2000) pp. 318-319 or Fisher et al. (2001) pp. 679-681.

stock results from producing in full batches, while demand usually does not have this pattern, i.e. demand is usually not in multiples of batches. This can be exploited by the addition of valid inequalities which adapt the ideas of (6-3) and tighten them (6-31).

$$I_{jt} \geq \left(bs_j \cdot \left\lceil \sum_{s=1}^{t+p} \frac{d_{js}}{bs_j} \right\rceil - \sum_{s=1}^{t} d_{js} \right) \cdot (1 - Z_{jt} - \sum_{s=1}^{p} Y_{jt+s}) \quad \forall j \in \mathcal{J}, t=1..T, p=0..T\text{-}t \quad (6\text{-}31)$$

In the first parenthesis the amount of product j is calculated that has to reside in stock at the end of period t, if there is no production of j in the interval $[t+1;t+p]$. This is done by adding up demands of periods 1 to $t+p$ and rounding to the next bigger integer number of full batches that have to be produced to fulfill demands up to period $t+p$ and subtracting the demands already satisfied. Together with the second parenthesis a lower bound is derived for inventory variables I_{jt}.

In the presence of batch flow restrictions (6-31) can be tightened even further for p=0. Then, at the end of period t some inventory must reside in stock, if the demand up to period t is not an integer multiple of the batch size. The difference between the number of full batches minimally produced up to t and the demand up to period t is a lower bound on inventory at the end of period t. This is expressed by constraints (6-32) which can be added to the model formulation if batch flow restrictions are present.

$$I_{jt} \geq \left(bs_j \cdot \left\lceil \sum_{s=1}^{t} \frac{d_{js}}{bs_j} \right\rceil - \sum_{s=1}^{t} d_{js} \right) \quad \forall j \in \mathcal{J}, t=1..T \quad (6\text{-}32)$$

A better way of accomplishing the same result is not to increase the model size by these additional constraints, but to preprocess the problem data such that these additional constraints are redundant. In presence of batch flow restrictions this can be achieved by modifying the demand data. Knowing that overproduction is necessary because full batches have to be completed before the batch can be used to fulfill demand, it is possible to shift portions of demands to earlier periods.[304] This shifting of demands should result in preprocessed demands (denoted by $d_{jt}^{(pre)}$) which are integer multiples of the batch size and deviate by the originally demands minimally. The formula for this shifting of demands to earlier periods is given by (6-33).

$$d_{jt}^{(pre)} = bs_j \cdot \left\lceil \sum_{s=1}^{t} \frac{d_{js}}{bs_j} \right\rceil - bs_j \cdot \left\lceil \sum_{s=1}^{t-1} \frac{d_{js}}{bs_j} \right\rceil \quad \forall j \in \mathcal{J}, t=1..T \quad (6\text{-}33)$$

This shift of demands will not alter the optimal solution in terms of the production decisions taken, but the resulting objective function value will decrease. The reason for this is that less inventory holding costs are attributed to the optimal solution, because demands occur earlier in time. This difference can be calculated by

304 A similar procedure is described by Fleischmann (1994, p. 396) and Salomon et al. (1991, p. 806) for the DLSP. There production is not in multiples of a batch size, but in multiples of a period's capacity. Consequently, demands are preprocessed to show this characteristic.

(6-34) from the difference in time the demands occur in the original data compared to the preprocessed data.

$$COST^{(pre)} = \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} h_j \cdot (\sum_{s=1}^{t} d_{js}^{(pre)} - \sum_{s=1}^{t} d_{js}) \quad (6\text{-}34)$$

Consecutive Campaigns of the same product

It has been argued above, that to allow for two (or more) consecutive campaigns of the same product is a desirable attribute of the model formulation if maximal campaign length restrictions are present. Due to (6-28) these are not allowed by the model formulation presented above.

A straightforward extension of the model formulation that allows for two (or more) consecutive campaigns of the same product would be to double the number of variables in such a way that there are two variables of each kind for each product. These two sets of variables can then be used alternately to model campaigns. The only difference between this modified model formulation and the one proposed above would be its model size and a slight adaptation of the inventory balance constraints (2-21), in which the two production amount variables X_{jt} of each product would need to be joined. To enhance computational performance it seems wise to remove symmetry[305] from the new model formulation by adding constraints (6-35).

$$Y_{J+j,t} \leq Z_{jt-1} \qquad \forall j=1..J,\ t \in \mathcal{T} \quad (6\text{-}35)$$

In (6-35) it is assumed that the new variables have been indexed with $J+1..2 \cdot J$ for products. Consequently, these constraints allow only setups for this new set of variables, if the same product has been produced immediately before. Thereby, the second set of variables can only be used for the second (fourth, ...) consecutive production run of each product.

Nevertheless, this approach considerably enlarges the model size. Thus, a second modeling approach will be described next. This one uses far less additional variables, but requires several adjustments of the initial model formulation presented above.

The basic idea behind this second model formulation is to split the production amount variables X_{jt} into two distinct sets of variables X_{jt}^b and X_{jt}^e. The superscripts b and e denote if production takes place at the *b*eginning or at the *e*nd of a certain period. Thereby, the number of (continuous) variables of the model formulation is increased by $J \cdot T$.

Production amount variables X_{jt} are used in the inventory balance constraints (2-21), capacity constraints (2-22) and setup constraints (6-2) of the basic PLSP. In the former two, X_{jt} is simply replaced by the sum of X_{jt}^b and X_{jt}^e. With respect to setup constraints a disaggregation is possible ((6-36) and (6-37)).

---

[305] Sherali and Smith (2001) show the advantages of removing symmetry for various model formulations. Cf. Sherali and Smith (2001) pp. 1396-1407.

$$X_{jt}^{b} \leq \min\left\{\frac{c_t}{a_j}, \sum_{\substack{s\in\mathcal{T}\\ s\geq t}} d_{js}\right\}\cdot Z_{jt-1} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}36)$$

$$X_{jt}^{e} \leq \min\left\{\frac{c_t}{a_j}, \sum_{\substack{s\in\mathcal{T}\\ s\geq t}} d_{js}\right\}\cdot Y_{jt} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}37)$$

Variables:

X_{jt}^{b} Production quantity of item j at the beginning of period t (first campaign in t)

X_{jt}^{e} Production quantity of item j at the end of period t (second campaign in t)

Production at the beginning of a period is possible only if the setup state has been carried over from the last period (6-36), whereas production at the end of a period is possible only if a setup operation has been performed in this period (6-37).

This split of production amount variables becomes necessary, because the production amount of one period may otherwise be attributed to two different campaigns. If there are two consecutive campaigns of the same product, the partitioning of the production amount to the beginning and to the end of a period resolves this issue. The exact partitioning is done arbitrary (by the solver) to reflect the needs of both campaigns and their associated restrictions.

Next, the defining constraints of campaign variables K_{jt} ((6-18)–(6-20)) need to be altered.

$$K_{jt} \leq K_{jt-1} + X_{jt}^{b} + maxlot_j \cdot Y_{jt} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}38)$$

$$K_{jt} \geq K_{jt-1} + X_{jt}^{b} - maxlot_j \cdot Y_{jt} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}39)$$

$$K_{jt} \leq X_{jt}^{e} + maxlot_j \cdot (1 - Y_{jt}) \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}40)$$

$$K_{jt} \geq X_{jt}^{e} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}41)$$

Constraints (6-38) and (6-39) replace (6-18) and (6-19). If no setup operation of product j in period t is performed (Y_{jt}=0), production quantities are accumulated in campaign variables K_{jt}. The two constraints provide (identical) upper and lower bounds in this case. Otherwise, if a setup operation is performed (Y_{jt}=1), both constraints do not restrict the allowed domain of campaign variables K_{jt}.

Here, campaign variables K_{jt} are initialized by (6-40) and (6-41) (which replace (6-20)) each time a setup operation for j is performed in t. In this case, (6-40) and (6-41) provide (identical) upper and lower bounds. In contrast to (6-20), in which campaign variables K_{jt} are initialized to zero in the period preceding a new setup operation, here campaign variables K_{jt} are initialized to the initial production amount in the starting period of a production run. Due to (6-37) this is the only period of one production run (campaign) in which production amount variables X^e_{jt} take values other than zero. In contrast to (6-21), variables K_{j0} are initialized at the beginning of the planning interval as $minlot_j$.

After having defined campaign variables K_{jt} by means of constraints (6-38)–(6-41), the restrictions regarding minimal lot sizes, maximal lot sizes and lot sizes requiring to be multiples of a batch size have to be derived. This is done here quite similar to the model formulation presented above ((6-22)–(6-25)). In contrast to the initial model formulation, in which the production amount of the current campaign of product j in period t is given by the sum of variables K_{jt} and X_{jt}, here, the production amount of the current campaign of product j in period t is given by the sum of variables K_{jt} and X^b_{jt}. Furthermore, as consecutive campaigns of the same product are now allowed, the index ranges in the sums of constraints (6-22) and (6-25) now also include product j. Both changes lead to constraints (6-42)–(6-45) replacing (6-22)–(6-25).

$$K_{jt-1} + X^b_{jt} \geq minlot_j \cdot \sum_{k \in \mathcal{J}} Y_{kt} \qquad \forall j \in \mathcal{J},\ t \in \mathcal{T} \quad (6\text{-}42)$$

$$K_{jt-1} + X^b_{jt} \leq maxlot_j \qquad \forall j \in \mathcal{J},\ t \in \mathcal{T} \setminus \{1\} \quad (6\text{-}43)$$

$$K_{jt-1} + X^b_{jt} = bs_j \cdot R_{jt} + S_{jt} \qquad \forall j \in \mathcal{J},\ t \in \mathcal{T} \setminus \{1\} \quad (6\text{-}44)$$

$$S_{jt} \leq bs_j \cdot \left(1 - \sum_{k \in \mathcal{J}} Y_{kt}\right) \qquad \forall j \in \mathcal{J},\ t \in \mathcal{T} \setminus \{1\} \quad (6\text{-}45)$$

As in the initial model formulation the different restrictions regarding the campaign size can be used rather independently.

However, a downside of this model formulation is that setup operation variables Y_{jt} need to be defined as binary variables here. This was not necessary in the initial model formulation, because due to constraints (2-25), (6-1), (6-4) and (6-28) they took only binary values in feasible solutions. Here, constraints (6-28) are no longer valid, because they forbid two consecutive setup operations of the same product. Moreover, constraints (6-4) are no longer valid inequalities, but become necessary constraints here.

6.1.2.2 Basic Model: CLSPL

The same modeling trick that worked to enhance the PLSP to allow for consecutive campaigns of the same product almost suffices to enhance the CLSPL to allow for lot sizes spanning over several periods. As has been argued above, a prerequisite for such a model formulation is that the setup state at each period boundary is uniquely defined.[306] This is done by variables W_{jt} in the CLSPL model formulation consisting of objective function (2-1) and constraints (2-2), (2-3), (2-5), (2-29), (2-32), (2-34), (6-16) and (6-17). The link variables W_{jt} of the CLSPL indicate the setup state at the beginning of a certain period t and corre-

[306] See section 6.1.1.2.

spond to variables Z_{jt-1} of the PLSP which indicate the setup state at the end of period t–1 (which obviously matches the setup state at the beginning of period t).

In solutions of the CLSPL it is possible that production of a certain product takes place at the beginning and at the end of a certain period, but some other product(s) are produced in between.[307] Therefore it is necessary to split production into two variables X^b_{jt} and X^e_{jt} as has been done above for the PLSP with consecutive campaigns of the same product. Again, the sum of these variables replaces production variables X_{jt} in the inventory constraints (2-2) and capacity constraints (2-3), whereas the setup constraints (2-32) are altered accordingly to (6-36) and (6-37).

$$X^b_{jt} \leq \min\left\{\frac{c_t}{a_j}, \sum_{\substack{s\in\mathcal{T} \\ s\geq t}} d_{js}\right\} \cdot W_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \quad (6\text{-}46)$$

$$X^e_{jt} \leq \min\left\{\frac{c_t}{a_j}, \sum_{\substack{s\in\mathcal{T} \\ s\geq t}} d_{js}\right\} \cdot Y_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \quad (6\text{-}47)$$

The defining constraints for campaign variables K_{jt} ((6-21) and (6-38)–(6-41)) can be taken directly from the model formulation for the PLSP with consecutive campaigns of the same product. Only for checking whether minimal, maximal or batch size restrictions are obeyed, slight adaptations to (6-42)–(6-45) are necessary. These concern the term that sums up variables Y_{jt} in constraints (6-42) and (6-45). This is not a valid method for the CLSPL, because in contrast to the PLSP, here generally more than one of these variables can take the value "1" in a certain period.

In order to check minimal lot sizes and batch size restrictions the information is needed whether *any* setup operation occurs in period t. Therefore, new continuous variables YI_t are introduced to indicate if any setup operation takes place in period t or not. These variables are defined by (6-48) and (6-49) and are used to replace the term $\sum_{k\in\mathcal{J}} Y_{kt}$ in constraints (6-42) and (6-45).

$$YI_t \geq Y_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \quad (6\text{-}48)$$

$$YI_t \leq \sum_{j\in\mathcal{J}} Y_{jt} \qquad \forall\, t\in\mathcal{T} \quad (6\text{-}49)$$

Variables:

YI_t Setup operation indicator for period t (=1, if a setup operation occurs in period t, =0 otherwise)

By definition, variable YI_t will be forced to zero, if no setup operation occurs in period t (6-49). On the other hand, if any setup operation occurs, (6-48) forces YI_t to become at least one. Furthermore, modified constraints (6-45) ensure that YI_t do not exceed this value (due to the non-negativity of variables S_{jt}). Thereby, vari-

[307] See e.g. the example by Suerie and Stadtler (2003) p. 1053.

ables YI_t take only binary values although they are defined as continuous variables.

Summarizing, due to the analogy of variables Z_{jt-1} of the PLSP and W_{jt} of the CLSPL only minor modifications are necessary to transfer the model formulation for lot sizes that span over several periods from PLSP to CLSPL.

More Simple Formulations – Campaigns within periods

Although not explored in more detail, for the sake of completeness also more simple model formulations for the aforementioned restrictions will be presented briefly. These apply only if the period length is considerably larger than the expected lot size and therefore the representation defect does not grow too big if the effect of period overlapping lot sizes is neglected.

In this case, the CLSP[308] can serve as a basic model ((2-1)–(2-6)). Minimal lot sizes are introduced by defining a lower bound on production, whenever production takes place or by defining variables X_{jt} as semi-continuous.

$$X_{jt} \geq minlot_j \cdot Y_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (6\text{-}50)$$

The maximal lot size is observed, if the big number b_{jt} limiting production in each period in (2-4) is replaced by $maxlot_j$. Last, to facilitate batch production, production quantity variables X_{jt} might simply be replaced by $bs_j \cdot R_{jt}$ with R_{jt} being integer.

Again, these much simpler measures do not help to overcome the representation defect in any sense, but are a much simpler form to deal with these kind of restrictions (minimal lot sizes, maximal lot sizes, batch restrictions), if less degree of accuracy is justifiable.

6.1.3 Time Continuity – Setup Operations

In this section period overlapping setup operations (setup times) will be considered. First, the PLSP is chosen as a basic model, followed by modifications which show the necessary adaptations if the CLSPL serves as a basic model.

6.1.3.1 Basic Model: PLSP

With the PLSP[309] chosen as a basic model, two alternative extensions to allow for period overlapping setup times will be introduced. The first one (called *POST*1) rests on the idea of cumulating setup times in analogy to the model formulation presented above for period overlapping lot sizes, where those have been cumulated in successive periods, if they belonged to one lot. The second one (called *POST*2) builds on a different modeling idea, which needs less variables (binary and continuous) and less constraints than the first one, but has some mild assump-

---

308 See section 2.2.

309 See section 2.3.3.

tions on the periods' lengths. After both alternatives will have been presented, some valid inequalities will be proposed.

The basic PLSP consists of objective function (2-20) and constraints (2-21), (2-22), (2-25)–(2-27), (6-1) and (6-2). Inequalities (6-3) remain valid. This is not true for inequalities (6-4), because with period overlapping setup times included in the model formulation, the modeling capabilities in terms of products that are "active" in a period are increased:

In the basic PLSP it is allowed to set up one product in each period. This means, that in each period there are at most two products active, one at the beginning of the period and one at the end of the period, whereby the latter has been set up during the period (see also Fig. 6-1, basic PLSP). With period overlapping setup times (*POST*) a third product may become active during a certain period. In this case, again the first product is produced at the beginning of the period, the second product is set up during the period and produced thereafter and at the end of the period the setup operation for the third product is started (see also Fig. 6-1, *POST*, period t=2). It might well be the case that the total setup time for this third product is attributed to this period, but due to the restriction of the basic model, that at most one setup is *finished* in each period, the setup of this third product can only be completed in the next period.

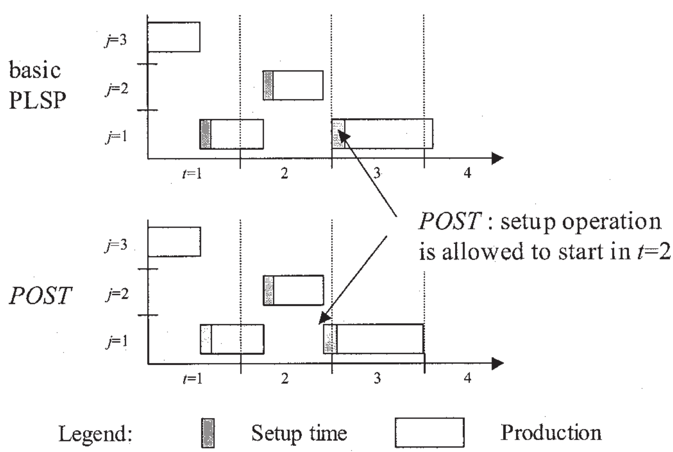


Fig. 6-1: Illustration of modeling capabilities of model formulations *POST* compared to the basic PLSP.

Period Overlapping Setup Times Variant 1 (*POST*1)

Model formulation *POST*1 needs three new types of variables, one of them being binary. The first type of variables, ST_{jt}, contains the setup time attributed to a setup operation of product j in period t. Variables ST_{jt} are continuous variables. Furthermore, variables KS_{jt} are defined to contain the cumulated setup time of the current setup for product j up to period t. They correspond to variables K_{jt} of the above model formulation for period overlapping lot sizes.[310] Finally, binary variables ZS_{jt} need to be defined. In a sense, these correspond to setup state variables Z_{jt}. Whereas variables Z_{jt} are equal to "1", if the resource is set up for product j at the end of period t and production can continue in t+1, variables ZS_{jt} express the same with respect to the setup operation. They become "1", if at the end of period t a setup operation for product j is going on and can continue in the following period t+1.

With these definitions in mind, model formulation *POST*1 (extending the basic PLSP[311]) can be stated as follows.

$$\sum_{j\in\mathcal{J}} a_j \cdot X_{jt} + \sum_{j\in\mathcal{J}} ST_{jt} \le c_t \qquad \forall\, t\in\mathcal{T} \quad (6\text{-}51)$$

$$\sum_{j\in\mathcal{J}} Z_{jt} + \sum_{j\in\mathcal{J}} ZS_{jt} = 1 \qquad \forall\, t\in\mathcal{T} \quad (6\text{-}52)$$

$$X_{jt} \le \min\left\{\frac{c_t}{a_j}, \sum_{\substack{s\in\mathcal{T}\\ s\ge t}} d_{js}\right\} \cdot (Z_{jt-1} + Y_{jt}) \qquad \forall\, j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}53)$$

$$KS_{jt-1} + \frac{1}{st_j} \cdot ST_{jt} = Y_{jt} + KS_{jt} \qquad \forall\, j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}54)$$

$$KS_{jt} \le 1 - \sum_{k\in\mathcal{J}} Y_{kt} + \frac{1}{st_j} \cdot ST_{jt} \qquad \forall\, j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}55)$$

$$KS_{jt} \le ZS_{jt} \qquad \forall\, j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}56)$$

$$Z_{jt} \le 1 - \sum_{\substack{k\in\mathcal{J}\\ k\ne j}} Y_{kt} \qquad \forall\, j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}57)$$

$$ST_{jt} \le c_t \cdot \left(Y_{jt} + ZS_{jt}\right) \qquad \forall\, j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}58)$$

$$ST_{jt} \ge 0, \quad KS_{jt} \ge 0 \qquad (KS_{j0} = 0) \qquad \forall\, j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}59)$$

$$Y_{jt}, Z_{jt}, ZS_{jt} \in \{0;1\} \qquad (Z_{j0} = ZS_{j0} = 0) \qquad \forall\, j\in\mathcal{J}, t\in\mathcal{T} \quad (6\text{-}60)$$

Variables:

KS_{jt}	Cumulated setup time of the current setup operation for product j up to period t [in %]
ST_{jt}	Setup time attributed to a setup of product j in period t
ZS_{jt}	Binary setup operation state variable (=1, if a setup operation for product j is going on at the end of period t (and can continue in period t+1); =0 otherwise)

---

310 See section 6.1.2.1.

311 The basic PLSP consists of objective function (2-20) and constraints (2-21), (2-22), (2-25)–(2-27), (6-1) and (6-2).

Constraints (6-51) are capacity constraints. They replace constraints (2-22) of the basic model. Here, the second term on the LHS has changed to take into account that the setup time is possibly split over two (or more) periods. Consequently, only the setup time attributed to this period is deducted from available capacity.

Constraints (6-1) of the basic model are supplemented by a second term on the LHS to yield (6-52). These constraints identify the setup state at each period boundary. Either one of the variables Z_{jt} equals "1", indicating that production of product j is possible at the end of period t and at the beginning of t+1, or one of the variables ZS_{jt} equals "1", indicating that a setup operation of product j can continue from the end of period t to the beginning of t+1.

Constraints (6-53) replace (6-2) of the basic model. These constraints restrict production of a certain product j to periods t, in which either the setup state is carried over from the preceding period ($Z_{jt-1}=1$) or a setup operation is finished ($Y_{jt}=1$).

The set of constraints (6-54)–(6-56) is needed to cumulate setup times. First, in (6-54) setup times ST_{jt} are cumulated into variables KS_{jt}. Once enough setup time is attributed to a setup operation ($st_j \cdot KS_{jt} + ST_{jt} \geq st_j$), binary variables Y_{jt}, which indicate the completion of a setup operation, can become "1". Finally, variables KS_{jt} are reset to the setup time of the current period, if a setup operation for any product has been completed in period t (6-55). To avoid that setup times for more than one product are accumulated, (6-56) resets all but at most one cumulation variable KS_{jt} to zero. This suffices, because of (6-54) and (6-55) no "wrong" setup time can be accumulated. Apart from the last setup operation within the planning interval, the correct amount of setup time is attributed to each setup operation (due to (6-54) and (6-55)). Too much setup time can be attributed to the last setup operation, because at the end of the planning horizon KS_{jt} is not restricted, but this is avoided if capacity is scarce (and needs to be used otherwise), or the resource would sit idle instead.

Binary setup state variables Z_{jt} are reset to zero directly, whenever a setup operation for any product $k \neq j$ is completed due to constraints (6-57). But this may not suffice, because production must also cease, if any other setup operation is started. Consequently, constraints (6-58) force either the binary setup operation completion variables Y_{jt} or the binary setup operation state variables ZS_{jt} to "1", if a setup operation for product j is started (or continuing) in period t. This, together with constraints (6-52) helps to end any production run.

Finally, (6-59) and (6-60) impose non-negativity and binary conditions on the variables used.

Period Overlapping Setup Times Variant 2 (*POST2*)

The second variant (*POST2*) follows from a totally different modeling idea. Here, compared to the initial PLSP model formulation, only two additional types of variables (both continuous) need to be defined. These two types of variables (Y_{jt}^1 , Y_{jt}^2) are used to model setup operations (setup time). They give the relative share of the setup time for product j, which (w.l.o.g.) is either attributed to the end of period t (Y_{jt}^2) or to any point in time in period t (Y_{jt}^1).

The idea is to split the setup time into two components. The first component (Y_{jt}^1) holds that part of setup time resulting in the period the setup operation is completed in (e.g., in t). The remaining setup time must have been accrued in periods preceding t. Therefore, only this other component (Y_{jt}^2) is added up for a certain number of periods (sp_j–1) to yield the complete setup time st_j. Data sp_j indicates, over how many periods a setup operation for product j is maximally split. How to derive these sp_j in any given situation will be discussed later. Initially, we will assume a constant capacity c per period. In this case a setup time st_j can be split over $\lceil st_j / c \rceil$ or $\lceil st_j / c \rceil + 1$ periods. Thus, the maximum number of periods the setup time is split over is computed by the latter term.

Furthermore, the index range of variables Z_{jt} is extended by a dummy product state (j=0). These new variables are also continuous and will be used to reflect, that the resource is not set up for a certain product, but that a setup operation is going on at the period boundary between t and t+1. In the latter case, variables Z_{0t} will take the value one and otherwise zero.

With these definitions the basic PLSP model formulation[312] is replaced/ extended by the following constraints to yield model formulation *POST2*.

$$\sum_{j\in\mathcal{J}} a_j \cdot X_{jt} + \sum_{j\in\mathcal{J}} st_j \cdot \left(Y_{jt}^1 + Y_{jt}^2\right) \le c_t \qquad \forall\, t\in\mathcal{T} \quad (6\text{-}61)$$

$$\sum_{j=0}^{J} Z_{jt} = 1 \qquad \forall\, t\in\mathcal{T} \quad (6\text{-}62)$$

$$X_{jt} \le \min\left\{\frac{c_t}{a_j}, \sum_{\substack{s\in\mathcal{T}\\ s\ge t}} d_{js}\right\}\cdot (Z_{jt-1} + Y_{jt}) \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \quad (6\text{-}63)$$

$$Y_{jt} \le Y_{jt}^1 + \sum_{s=t+1-sp_j}^{t-1} Y_{js}^2 \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \quad (6\text{-}64)$$

$$\sum_{j\in\mathcal{J}} Y_{jt}^2 \le Z_{0t} \qquad \forall\, t\in\mathcal{T} \quad (6\text{-}65)$$

$$Z_{jt} \le 1 - \sum_{\substack{k\in\mathcal{J}\\ k\ne j}} Y_{kt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \quad (6\text{-}66)$$

$$Y_{jt} \le 1 - \sum_{\substack{k\in\mathcal{J}\\ k\ne j}} Y_{kt-1}^2 \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \quad (6\text{-}67)$$

$$\sum_{\substack{k\in\mathcal{J}\\ k\ne j}} Y_{kt} + \sum_{s=t}^{t+sp_j-3} Y_{js} \le 1 \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T},\ \text{if } sp_j\ge 4 \quad (6\text{-}68)$$

312 The basic PLSP consists of objective function (2-20) and constraints (2-21), (2-22), (2-25)–(2-27), (6-1) and (6-2).

$$Y_{jt}^1, Y_{jt}^2 \geq 0 \qquad \forall\, j \in \mathcal{J},\, t \in \mathcal{T} \quad (6\text{-}69)$$

$$Y_{jt}^2 = 0 \qquad \forall\, j \in \mathcal{J},\, t{=}2{-}sp_j..0 \quad (6\text{-}70)$$

$$Z_{0t} \geq 0 \qquad \forall\, t \in \mathcal{T} \quad (6\text{-}71)$$

$$Y_{jt}, Z_{jt} \in \{0;1\} \quad (Z_{j0} = 0) \qquad \forall\, j \in \mathcal{J},\, t \in \mathcal{T} \quad (6\text{-}72)$$

Data:

sp_j Maximum number of periods necessary for a setup operation for product j ($sp_j = \lceil st_j / c \rceil + 1$)

Variables:

Y_{jt}^1 Relative share of setup time for product j in period t (start or within)
Y_{jt}^2 Relative share of setup time for product j in period t (end)
Z_{0t} Setup state variable, indicates whether any setup state persists at the end of period t (=0) or none (=1). The latter case indicates that a setup operation for a product is going on at the end of period t

Constraints (6-61) are capacity constraints which replace (2-22) of the original model formulation. Only the second term on the LHS is altered to reflect that setup times are possibly split over two (or more) periods. By the second term the share of the setup time attributed to the respective period is taken into account.

A unique setup state is identified at each period boundary by constraints (6-62) which replace (6-1) of the basic model formulation. Only the range of the summation on the LHS is changed to take the dummy product state (j=0) into account. Thereby, constraints (6-62) indicate, that either a setup operation is going on at the boundary of periods t and t+1 (Z_{0t}=1) or a setup state for any product $j \in \mathcal{J}$ persists at this point in time (Z_{jt}=1).

Similarly to (6-53) in model formulation *POST*1, constraints (6-63) replace (6-2) of the basic model. These constraints restrict production of a certain product j to periods t, in which either the setup state is carried over from the preceding period (Z_{jt-1}=1) or a setup operation is finished (Y_{jt}=1). As argued above, this allows that up to three products are active in each period, although only two of them are allowed to be produced, but idle time at the end may then be used to start a (second) setup operation (see also Fig. 6-1).

Constraints (6-64) specify that a setup operation for a certain product j in period t is only completed (Y_{jt}=1), if enough setup time has been reserved in the sp_j periods, the setup time has been maximally split over (t–sp_j+1 .. t). Then, the sum of variables representing the relative share of setup times in certain periods must equal at least "1" in this interval. Splitting of variables Y_{jt}^1 and Y_{jt}^2 prevents that a fraction of the setup time is attributed to more than one setup operation.

A setup state for a certain product at a period boundary is lost, whenever a setup operation is going on at the end of this period due to (6-65). If a setup operation is going on at the end of period t, indicated by a positive value of variable

Y_{jt}^2, no setup state can be saved (all Z_{jt}=0 for $j \in \mathcal{J}$), and due to (6-62) the dummy product state indicator (Z_{0t}) will become "1".

Constraints (6-66) are known from model formulation *POST*1 (6-57). Like constraints (6-65) they reset binary setup state variables Z_{jt} to zero, whenever a setup operation for any product $k \neq j$ is completed in period t.

Furthermore, (6-67) forbid completion of a setup operation in a certain period, if at the end of the preceding period a setup operation for any other product was going on. This is a valid constraint, because if at the end of period t–1 a setup operation for product $k \neq j$ was going on, this setup operation would have to continue and finish before a setup operation for j would be completed. As both setup operations cannot be completed in t, the constraint is valid.

Only, for situations with $sp_j \geq 4$ another type of constraints (6-68) must be added to the model formulation. These constraints prevent that during long setup operations ($sp_j \geq 4$) one short setup operation is scheduled in between. Fig. 6-2 illustrates this special case. This situation arises, because the number of periods the setup operation is split over is not known in advance, but may be either sp_j or $sp_j - 1$. This means that in some cases one period within the interval $t - sp_j + 1$ .. t might be empty (here: t=2), therefore not imposing any restrictions on setup operation completion variables Y_{kt} for the other products $k \neq j$ in the subsequent period (here: t=3). This can be used by the solver to schedule a short production run for product k (here: k=2) within the interval the setup operation for j=1 should take place.

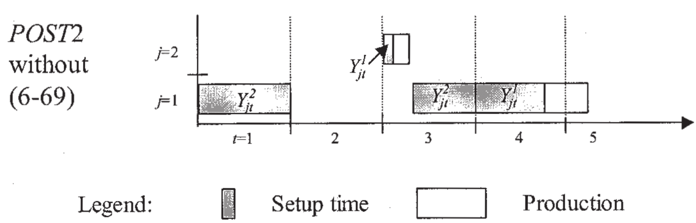


Fig. 6-2: Illustration of modeling error for $sp_j \geq 4$ if (6-68) is omitted in the model formulation *POST*2.

Constraints (6-68) prevent this special case. If a setup operation for product j begins in period t and can be split over maximally sp_j periods than its earliest completion will be in $t + sp_j - 2$. Thus, if a setup operation is completed for any product $k \neq j$ in period t, no setup operation for product j can be finished in the interval $t..t + sp_j - 3$. As it is also impossible that two setup operations for product j are completed within such a short interval ($t..t + sp_j - 3$), constraints (6-68) are valid constraints.

Finally, (6-69)–(6-72) impose non-negativity and binary conditions on the variables used.

Derivation of assumptions on sp_j

Two sets of constraints ((6-64) and (6-68)) make use of data sp_j. In (6-64) fractions of the setup time are added up, whereas in (6-68) the completion of two setup operations is forbidden, if these were too close together. This concept works fine as long as capacity is assumed constant in each period.

The latent danger of failing inherent to model formulation *POST2* lies in the fact that it is not known a priori, whether the setup time is split over sp_j or sp_j-1 periods. Whereas in model formulation *POST1* the cumulated setup time is reset to zero as soon as a setup operation is completed, the danger here is that attributing a fraction of setup time of a certain period to two different setup operations might become possible. The reason for this is that the summation in (6-64) must assume that the setup time is split over sp_j periods. To avoid double counting of setup times, setup time variables have been split into two variables Y_{jt}^1 and Y_{jt}^2. Fig. 6-3 illustrates this case. The setup operation for j=1 which is finished in t=6 could have been split over three periods (4..6) like in the example shown here or over sp_j=4 periods (3..6). This is why (6-64) must add up the setup time variables of periods 3..6. Attributing the setup time for product j=1 in period t=3 to both setup operations (the one finished in t=3 and the one finished in t=6) is avoided by distinguishing between setup times that lie at the beginning of a period and those that lie at the end, which is done by splitting the setup time into variables Y_{jt}^1 and Y_{jt}^2.

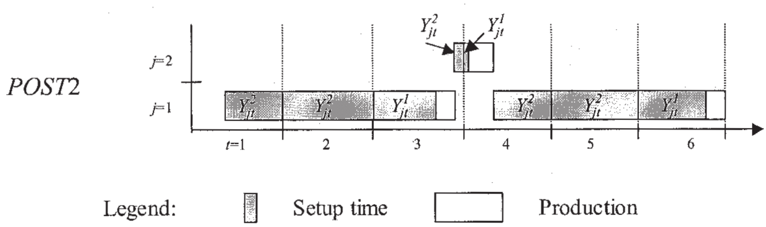


Fig. 6-3: Assignment of setup time variables in model formulations *POST2*.

Generally speaking, this distinction suffices to allocate setup times correctly as long as the number of periods the setup time is eventually split over differs only by one (sp_j and sp_j-1).

Now the assumption that capacity is constant will be lifted. Consequently, data sp_j cannot be determined as a constant over time either, because as capacities vary, the number of periods the setup time may be split over also varies. Therefore, we define sp_{jt} as the number of periods the setup time is maximally split over, if the setup operation of product j is finished in period t. sp_{jt} evaluates to

$$sp_{jt} = 1 + \min\left\{ n \mid st_j \leq \sum_{s=1}^{n} c_{t-s} \right\} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (6\text{-}73)$$

The two affected constraints are aligned as follows. In (6-64) sp_j is simply replaced by sp_{jt}, whereas in (6-68) the second sum must consider that the decision, if a setup operation must have started in t, is not dependent on sp_{jt}, but rather on sp_{js}, with s being the period the setup operation is potentially finished in. This is taken into account in (6-74).

$$\sum_{\substack{k\in\mathcal{J}\\k\neq j}} Y_{kt} + \sum_{\substack{s=t,if\\s-sp_{js}+2<t}}^{T} Y_{js} \leq 1 \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \quad (6\text{-}74)$$

Two cases need to be distinguished:

- $c_t \geq c_{t+1}$ (capacities are constant or decreasing over time)
  When capacities are constant or decreasing over time, one can show that setup times are split over either sp_{jt} or $sp_{jt}-1$ periods. These are cases the above model formulation is valid for.
  Assuming that sp_{jt} has been calculated by (6-73), the following inequality $0 < st_j - c_{t-1} - c_{t-2} - \ldots - c_{t-n+1} \leq c_{t-n}$ holds. If $st_j - c_{t-1} - c_{t-2} - \ldots - c_{t-n+1} \leq c_t$ then the setup operation can be split either over $sp_{jt}-1$ or sp_{jt} periods, because the capacity in period $t-n$ can be used for the setup operation but does not have to be used. On the other hand, if $c_t < st_j - c_{t-1} - c_{t-2} - \ldots - c_{t-n+1}$ the whole setup operation cannot be completed in periods $t-n+1..t$ and must have started in $t-n$. This second case means that a setup operation for product j which finishes in period t must have started in $t-n$, removing any uncertainty on its starting period.
- $c_t < c_{t+1}$ (capacities are increasing over time)
  Unfortunately the model formulation is no longer valid in this case. This will be illustrated by a short example (see Fig. 6-4). A setup operation that is finished in period t=6 can start either in t=3 (scenario 1) or t=5 (scenario 2). Consequently, sp_{jt} must assume the (worst) case (scenario 1) for the accumulation of setup times in (6-64). But then in scenario 2 this leads to the undesirable behavior that the setup time in period t=3 is counted twice and attributed to the first setup operation, which is finished in t=4, as well as to the second setup operation, which is finished in t=6.

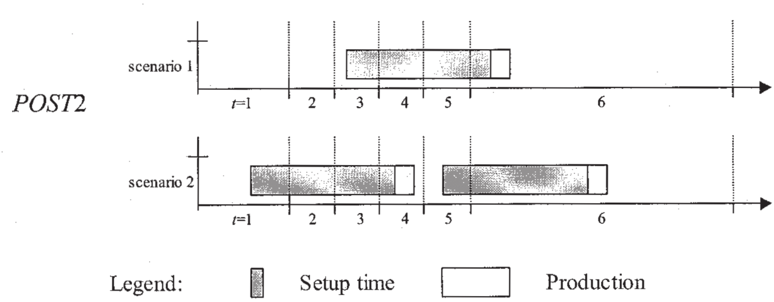


Fig. 6-4: Assignment of setup time variables in model formulations *POST2*.

Generally speaking, to distinguish between two setup time variables Y^1_{jt} and Y^2_{jt} is a legitimate measure as long as the number of periods setup times are eventually split over differs at most by one (sp_{jt} and $sp_{jt}-1$). If this is no longer the case, i.e. $c_t < c_{t+1}$ for any given t, model formulation *POST2* can fail.

Of course, one can avoid failing at the price of adding two further constraints similar to (6-54) and (6-55) of *POST1*. These constraints ((6-75) and (6-76)) ensure, that between two completions of setup operations a full setup time is scheduled. (6-75) take care that enough setup time is accumulated, while (6-76) reset cumulation variables KS_{jt} to zero. Furthermore, this construction makes (6-68) (rsp. (6-74)) obsolete, which have been used so far to tear two setup operations apart.

$$KS_{jt-1} + Y_{jt}^1 + Y_{jt-1}^2 = Y_{jt} + KS_{jt} \qquad \forall\, j \in \mathcal{J},\ t \in \mathcal{T} \quad (6\text{-}75)$$

$$KS_{jt} \le 1 - \sum_{k \in \mathcal{J}} Y_{kt} \qquad \forall\, j \in \mathcal{J},\ t \in \mathcal{T} \quad (6\text{-}76)$$

Following these arguments, one can conclude that model formulation *POST2* either requires the assumption that capacities are non-increasing over time or the integration of several additional constraints. On the other hand, no such assumption is necessary for model formulation *POST1*.

Valid Inequalities

From constraints (6-57) of model formulation *POST1* (which are the same as constraints (6-66) of model formulation *POST2*) it is possible to generate a set of valid inequalities. Constraints (6-57) and (6-66) express that if a setup operation for any product $k \neq j$ is completed in period t, it is impossible to have a setup state of product j at the end of t. This is obvious, because to move to setup state j at the end of period t would require a second setup operation (that for j) in addition to the one for $k \neq j$, which is not possible due to the assumptions of the basic PLSP model formulation.

Valid inequalities (6-77) turn this argument around. If there is a setup state for any product $k \neq j$ at the end of period t, a setup operation for j has not been completed in t.

$$\sum_{\substack{k \in \mathcal{J} \\ k \neq j}} Z_{kt} \le 1 - Y_{jt} \qquad \forall\, j \in \mathcal{J},\ t \in \mathcal{T} \quad (6\text{-}77)$$

These valid inequalities can be generalized. If two types of constraints like (6-78) and (6-79) with binary variables α_j and β_j are present, valid inequalities of the form (6-81) can be generated. The reasoning is as follows. If all binary variables β_j are "0", (6-81) is valid because of (6-78). If one variable β_j is "1", constraints (6-79) force all but one variable α_j to "0". The one variable α_j which is not forced to "0" has the same index as the variable β_j that was assumed to be "1". This is exactly, what is expressed by (6-81). The variable β_j, which evaluates to "1", forces any α_j with a different index $k \neq j$ to "0". As all the other β_j are "0" the remaining α_j is restricted to be less than or equal 1, which is valid due to (6-78). Finally, no more than one β_j can evaluate to "1", because otherwise (6-79) would not be feasible. Furthermore, constraints (6-81) render (6-78) redundant.

$$\sum_{j \in \mathcal{J}} \alpha_j \leq 1 \qquad (6\text{-}78)$$

$$\alpha_j \leq 1 - \sum_{\substack{k \in \mathcal{J} \\ k \neq j}} \beta_k \qquad \forall j \in \mathcal{J} \quad (6\text{-}79)$$

$$\alpha_j, \beta_j \in \{0;1\} \qquad \forall j \in \mathcal{J} \quad (6\text{-}80)$$

$$\sum_{\substack{k \in \mathcal{J} \\ k \neq j}} \alpha_k \leq 1 - \beta_j \qquad \forall j \in \mathcal{J} \quad (6\text{-}81)$$

In model formulations *POST*1 and *POST*2 constraints (6-57), (6-66) and (6-67) are of the proposed form and can be used to derive valid inequalities as described above.

6.1.3.2 Basic Model: CLSPL

The concept of period overlapping setup times is also portable to the basic model CLSPL. As model formulation *POST*1 was evaluated to be the more general model formulation if the PLSP served as basic model, only this variant will be transferred to the CLSPL. Furthermore, some ideas are borrowed from the transfer of the lot size extension from the PLSP to the CLSPL.

The basic CLSPL model formulation consists of objective function (2-1) and constraints (2-2), (2-3), (2-5), (2-29), (2-32), (2-34), (6-16), (6-17) and valid inequalities (6-11). Setup time variables ST_{jt} of model formulation *POST*1 will be split into two sets of variables ST_{jt}^{b} and ST_{jt}^{e}. The reason for this is that setup time for product *j* in period *t* might belong to two different setup operations, one at the beginning or some time inside the period (ST_{jt}^{b}), which is completed in period *t*, and one that is started and still continuing at the end of *t* (ST_{jt}^{e}). Still, the condition will be respected that in each period at most one setup operation for each product is completed.

With this definition made the basic CLSPL model formulation is replaced/extended by the following constraints to yield a model formulation which allows for period overlapping setup times.

$$\sum_{j \in \mathcal{J}} a_j \cdot X_{jt} + \sum_{j \in \mathcal{J}} (ST_{jt}^{b} + ST_{jt}^{e}) \leq c_t \qquad \forall t \in \mathcal{T} \quad (6\text{-}82)$$

$$\sum_{j \in \mathcal{J}} W_{jt+1} + \sum_{j \in \mathcal{J}} ZS_{jt} = 1 \qquad \forall t \in \mathcal{T} \quad (6\text{-}83)$$

$$ST_{jt}^{b} \leq \min\{c_t, st_j\} \cdot Y_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (6\text{-}84)$$

$$ST_{jt}^{e} \leq \min\{c_t, st_j\} \cdot ZS_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (6\text{-}85)$$

$$X_{jt} \leq \min\left\{\frac{c_t}{a_j}, \sum_{\substack{s \in \mathcal{T} \\ s \geq t}} d_{js}\right\} \cdot (W_{jt} + Y_{jt}) \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (6\text{-}86)$$

$$KS_{jt-1} + \frac{1}{st_j} \cdot \left(ST_{jt}^b + ST_{jt}^e\right) = Y_{jt} + KS_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (6\text{-}87)$$

$$KS_{jt} \leq 1 - YI_t + \frac{1}{st_j} \cdot ST_{jt}^e \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (6\text{-}88)$$

$$KS_{jt} \leq ZS_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (6\text{-}89)$$

$$ST_{jt}^b, ST_{jt}^e \geq 0, \quad YI_{jt} \geq 0, \quad KS_{jt} \geq 0, \quad (KS_{j0} = 0) \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (6\text{-}90)$$

$$Y_{jt}, Z_{jt}, ZS_{jt} \in \{0;1\} \quad (Z_{j0} = ZS_{j0} = 0) \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad (6\text{-}91)$$

Variables:

ST_{jt}^b Setup time attributed to a setup operation of product j at the beginning or somewhere in period t

ST_{jt}^e Setup time attributed to a setup operation of product j at the end of period t

Constraints (6-82) are capacity constraints and replace (2-3) of the basic model formulation. The difference is that setup times are not accounted for by binary variables, but by the actual amount of setup time attributed to the specific period which need not match a full setup time.

The state of the resource at the end of each period t is determined by (6-83), which replace (6-16) of the basic model formulation. Either a setup state has been reached in one of the preceding periods and production can continue in t+1 (one W_{jt}=1) or a setup operation is going on (one ZS_{jt}=1).

Constraints (6-84) and (6-85) define variables ST_{jt}^b and ST_{jt}^e. Only if a setup operation is completed in period t, variables ST_{jt}^b are allowed to become positive values, whereas variables ST_{jt}^e are allowed to become positive values only if a setup operation is still continuing at the end of period t.

Production of product j in period t is only allowed, if the corresponding setup state persists at the beginning of period t or a setup operation for product j is completed in period t (6-86).

Constraints (6-87)–(6-89) accumulate setup times like (6-54)–(6-56). First, in (6-87) setup times are cumulated into variables KS_{jt}. Once enough setup time is attributed to a setup operation, binary setup operation completion variables Y_{jt} can become "1". Finally, variables KS_{jt} are reset to the setup time at the end of the current period, if a setup operation for any product has been completed in period t (6-88). Thus, a setup operation for any product j can be completed within period t and a new setup operation for this product can start at the end of t, while several other products k≠j might have been produced in between. This case depicts the typical big-bucket scenario of the CLSPL, that even within a period several setup operations may be completed.

Finally, (6-90) and (6-91) impose conditions on the domains of variables. Of course, variables can also be initialized according to the initial plant state at the beginning of the planning interval.

6.1.4 Time Continuity – Resource Utilization

As has been argued in Section 3.5 to maintain a certain resource utilization is an issue in the process industries. Therefore, several additional constraints will be proposed to keep production at a certain level or shut down resources when necessary. Two scenarios will be distinguished. In the first scenario resources are allowed to be shut down, whereas in the second scenario, resources cannot be switched off safely and therefore have to keep a certain production level all the time. Furthermore, it is assumed that the setup state is lost, if a resource is switched off. The minimum utilization rate $util_j$ will be defined product dependent here.

6.1.4.1 Resources with Off Times

If resources are allowed to be switched off, it is necessary to add a product to the model formulation which represents idle time (j=–1). Thereby, it is possible to model idle periods, which may emerge. Other than that, the idle product is not necessary. Especially, the idle product need not be set up in a "regular" PLSP period, in which some product j is produced at the beginning, a setup operation is performed for product k and this product is then produced until the end of the period. In this case, idle time can be attributed to the time interval between the end of production of j and the beginning of the setup operation of k, such that both products (j and k) maintain their respective minimum utilization levels.

Basic Model PLSP and PLSP with Lot Size Extension

Two variants of modeling minimum resource utilization constraints for resources with off times will be presented for the basic model PLSP as well as for its extension with period overlapping lot sizes. The first one relates production quantity variables X_{jt} to binary setup state variables Z_{jt}.

$$X_{jt} \geq util_j \cdot \frac{c_t}{a_j} \cdot (Z_{jt-1} + Z_{jt} - 1) \qquad \forall\, j \in \mathcal{J},\, t \in \mathcal{T} \quad (6\text{-}92)$$

Data:

$util_j$ Minimum utilization rate for product j

If production covers the whole period (Z_{jt-1}=1 and Z_{jt}=1), constraints (6-92) demand that the actual production X_{jt} is above the minimum utilization rate $util_j$. In the first and the last period of the production run (either Z_{jt}=1 or Z_{jt-1}=1) constraints (6-92) are relaxed. The relaxation can be made, because in the first and last period of the production run, the remaining time can be filled by a setup operation, production of another product and/or with some idle resource time (off time).

In contrast to this first variant which leads to $\|\mathcal{J}\|\cdot\|\mathcal{T}\|$ additional constraints, in the second variant it suffices to add $2\cdot\|\mathcal{T}\|$ additional constraints to the model formulation.

$$X_{-1,t} + \sum_{j \in \mathcal{J}} \frac{a_j}{util_j} \cdot X_{jt} \geq c_t \cdot (1 - \sum_{j \in \mathcal{J}} Y_{jt}) \qquad \forall\ t \in \mathcal{T} \quad (6\text{-}93)$$

$$X_{-1,t} \leq c_t \cdot Z_{-1,t} \qquad \forall\ t \in \mathcal{T} \quad (6\text{-}94)$$

In periods with no setup operation (all Y_{jt}=0) constraints (6-93) force either the production variable of the specific product j which is produced to be above its minimum resource utilization level, or idle product ($X_{-1,t}$) must be "produced". Again, this restriction is relaxed in the first and last period of a production run. These are periods, in which a setup operation takes place (one Y_{jt}=1). Constraints (6-94) allow production of the "idle" product (j=–1) only, if this product has been properly set up.

PLSP with *POST* and PLSP with Lot Size Extension and *POST*

It is fairly easy to broaden the extension proposed for the basic PLSP to the PLSP with period overlapping setup times as well as to a model formulation with both extensions, period overlapping lot sizes and period overlapping setup times. The latter model formulation will be proposed in section 6.1.5.

The difference between this extension and the extension proposed for the basic PLSP is caused by the fact that here, it does not suffice to relax constraints (6-93) whenever a setup operation takes place in a period. The reason for this is obvious. In model formulations with period overlapping setup times, the setup operation usually does not take place within a single period. Therefore, (6-93) have to be relaxed whenever some part of the setup operation takes place in a period. This is indicated either by setup time completion variables Y_{jt} like in the basic PLSP model or by setup state variables indicating that a setup operation is going on at the end of the period. These are variables ZS_{jt} in model formulation *POST*1 or variables Z_{0t} in model formulation *POST*2.

Consequently, to get a valid extension also in this case the parenthesis on the RHS of constraints (6-93) has to be supplemented by the term $-\sum_{j \in \mathcal{J}} ZS_{jt}$ for model formulation *POST*1 and $-Z_{0t}$ for model formulation *POST*2.

Basic model CLSPL and CLSPL with Lot Size Extension

In analogy to the basic model PLSP, the basic model CLSPL as well as its extension with period overlapping lot sizes may be extended to incorporate minimum resource utilization by two variants.

As resources are allowed to be switched off, what can be done whenever a setup operation takes place in a period (between the end of production of the preceding product and the setup operation), constraints (6-92) are altered to yield (6-95) to take the characteristics of the basic model CLSPL into account.

$$X_{jt} \geq util_j \cdot \frac{c_t}{a_j} \cdot \left(W_{jt} + W_{jt+1} - 1 - \sum_{k \in \mathcal{J}} Y_{kt} \right) \qquad \forall\ j \in \mathcal{J},\ t \in \mathcal{T} \quad (6\text{-}95)$$

Variables W_{jt} replace variables Z_{jt-1} and a fourth term is added in the parenthesis on the RHS. The reasoning is the same as above. Only if the same setup state prevails at the beginning and at the end of period t (W_{jt}=1 and W_{jt+1}=1) and no setup operation has taken place within period t (all Y_{jt}=0), product j is produced during the whole period t and must utilize the resource according to its minimum utilization rate $util_j$. Other than that, (6-95) are relaxed, because then any idle time can be attributed to the time interval in front of (one of) the setup operation(s).

The second variant ((6-93) and (6-94)) can be taken directly from the basic PLSP. Only variables $Z_{-1,t}$ need be replaced by corresponding variables $W_{-1,t}$. When variables YI_t are present in the model formulation, these can be used to save some coefficients in the matrix. These may then substitute the sum on the RHS of (6-93).

CLSPL with *POST* and CLSPL with Lot Size Extension and *POST*

The extension proposed for the PLSP with period overlapping setup times as well as for a model formulation with both extensions, period overlapping lot sizes and period overlapping setup times, is also valid if the CLSPL is chosen as the basic model.

As the setup operation usually does not take place within a single period here, (6-93) have to be relaxed whenever some part of the setup operation takes place in a period. This is indicated either by setup time completion variables Y_{jt} (or indicator variables YI_t) or by setup state variables indicating that a setup operation is going on at the end of the period (ZS_{jt}).

Consequently, in order to get a valid extension also in this case, the parenthesis on the RHS of constraints (6-93) has to be supplemented by the term $-\sum_{j\in\mathcal{J}} ZS_{jt}$.

6.1.4.2 Resources without Off Times

If resources are not allowed to be switched off, the integration of minimum resource utilization constraints is somewhat easier. In this case any capacity apart from the amount used up by setup operations has to be filled with production being at least on its minimum utilization level.

A single constraint for each period suffices to model this for the basic models PLSP and CLSPL as well as their respective extensions which allow for period overlapping lot sizes.

$$\sum_{j\in\mathcal{J}} \frac{a_j}{util_j} \cdot X_{jt} \geq c_t - \sum_{j\in\mathcal{J}} st_j \cdot Y_{jt} \qquad \forall\, t\in\mathcal{T} \quad (6\text{-}96)$$

On the RHS any setup time is deducted from available capacity, which is then used as a lower bound on the production quantity variables with respect to their minimum utilization level.

When period overlapping setup operations are possible, this does not suffice. The reason for this is that is has been modeled there that a setup operation is completed whenever enough setup time has been accumulated. It may well be the case that too much setup time will be attributed to a certain setup operation. This does not harm the solution in the basic case, but here it may prevent the model from maintaining minimum utilization levels. To avoid this, the following global constraints must be added to model formulation *POST*1 (6-97) or *POST*2 (6-98), respectively.

$$\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} st_j \cdot Y_{jt} = \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} ST_{jt} \qquad (6\text{-}97)$$

$$\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} Y_{jt} = \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} (Y_{jt}^1 + Y_{jt}^2) \qquad (6\text{-}98)$$

Both simply state that the total amount of setup time in the whole planning interval is in line with the number of setup operations performed within the planning interval.

Requiring minimum resource utilization levels of resources without off times should be done with great care.[313] In the model formulations proposed here, demand fulfillment is assumed. This means that any production in excess of demand is carried in inventory until the end of the planning interval is reached. Consequently, if a minimum resource utilization is required, the plan will be distorted by unnecessary setup operations (to avoid excess production by increasing setup times) or unnecessary production of the cheapest product (to avoid inventory holding costs) – whatever is less expensive. Thus, some penalty cost (on excess production) should be added in the objective function to guide the selection and quantity of products that are produced to fulfill the minimum resource utilization level.

6.1.5 Combinations

The different enhancements of the basic model formulations PLSP and CLSPL, which take into account lot sizes that are split over several periods or period overlapping setup times, have been proposed as independent building blocks. Therefore, they are easily combined, if both of these restrictions are present in a certain situation.

Model formulation *POST2* as an extension of the basic PLSP model does not allow consecutive lots of the same product. This is not necessary in a basic PLSP model, but becomes a prerequisite when maximum lot size restrictions are present. The feature is only included in the *POST1* extension (of both, the PLSP and the CLSPL). Consequently, this extension has to be used dependent on whether the PLSP or the CLSPL with period overlapping lot sizes is extended to allow also for period overlapping setup times.

Furthermore, also both of the basic models (PLSP and CLSPL) might be combined. As the proposed building blocks are similar to each other, a model consisting of periods with a PLSP character and those with a CLSPL character can also be combined. In this case it suffices to secure that at the intersection of both models the resource state (i.e. variables W_{jt} and Z_{jt}) is well-defined and matches both sides.

With all these extensions made, there is only one limit left that restricts the representation of all plans that are possible on a continuous time scale within this time-discretized setting. This last limitation stems from the fact, that independent from whether the PLSP or the CLSPL is chosen as the basic model, the period ca-

[313] Cf. Suerie (2004) p. 14.

pacity has to be chosen such, that at most one setup operation of each product is finished in each period in an optimal (continuous time) solution. $maxlot_j$ limits the production of product j for each lot. Consequently, it is not possible to produce more than $2 \cdot maxlot_j$ of product j in a single period.

6.1.6 Further Modeling Enhancements

Finally, another modeling trick will be introduced. In the process industries production is often not only restricted by constraints on masses but also by constraints on time. An example for this will be presented with some computational results in section 7.5. There, one has to distinguish between timing constraints (a minimal lot size is measured in days of production), but at the same time this production time does not translate to an exact mass amount. The reason for this is that production output of a certain processing step is not fixed, but can be varied between certain levels ($minrate_j$ and $maxrate_j$).

Modeling this scenario is easily accomplished within the proposed setting. The trick lies in distinguishing between production variables which represent the mass amount (usually X_{jt}) and production variables which represent the time a certain product is produced (XT_{jt}).

The latter variables (XT_{jt}) are used to replace corresponding variables (X_{jt}) in the capacity constraints of the model formulation. Then minimal lot size restrictions are imposed on these variables. Finally, the production time variables (XT_{jt}) must be transferred to the production mass variables (X_{jt}) as these are used to fulfill demands. This is done by (6-99).

$$minrate_j \cdot XT_{jt} \leq X_{jt} \leq maxrate_j \cdot XT_{jt} \qquad \forall\, j \in \mathcal{J},\, t \in \mathcal{T} \quad (6\text{-}99)$$

Data:

$maxrate_j$ Maximal production rate of product j
$minrate_j$ Minimal production rate of product j

Variables:

XT_{jt} Production time of item j in period t

6.2 Integration into a Decomposition Heuristic

In this section a temporal decomposition heuristic will be described. It aims at solving larger problems, for which solving a MIP model formulation gets too time consuming. As the main contribution of this thesis has been the proposal of a new modeling approach that allows for time continuity within a time-indexed setting, not an entirely new decomposition heuristic has been developed, but a rather successful approach for the MLCLSP by Stadtler (2003) has been adapted to work also with these models.

First, the heuristic by Stadtler (2003) is briefly sketched in the next section. The two following sections will then describe in more detail the two most important building blocks of the heuristic, the rolling scheme and measures taken for anticipation, as well as necessary adaptations for the models considered in this thesis.

6.2.1 Outline

The main obstacle, why MIP model formulations cannot be used to solve larger problems, is the immense number of binary and/or integer variables that necessitate a large branch-and-bound tree. As has been argued in section 5.2.2, the idea to overcome this obstacle is to decompose the overall problem into subproblems, each holding only a limited number of binary (integer) variables. Then, a tight model formulation can be used to solve these subproblems with a standard MIP solver.

The temporal decomposition heuristic by Stadtler (2003)[314] divides the planning horizon into three parts (see also Fig. 6-5). The most important part is the rolling window. Each rolling window is associated with one planning step. In each planning step the rolling window is moved several periods farther, until the planning horizon is reached in the last planning step. Binary variables are only present in the model formulation inside the rolling window, i.e. setup decisions are made in each planning step only for those periods that belong to the rolling window. Setup decisions belonging to periods preceding the rolling window have been made in preceding planning steps and have been fixed according to solutions obtained in these planning steps. In periods following the rolling window no setup decisions are made, although capacity lost due to setup operations in these periods can be accounted for (heuristically).

The main idea of Stadtler (2003) has been to use a tight reformulation of the MLCLSP (SPL reformulation) inside the rolling window, while a standard formulation (I&L) is used for the rest of the planning horizon. Thus, in each planning step a model formulation relating to the complete planning interval is solved. However, setup decisions are made in each planning step only for periods belonging to the rolling window. The time interval preceding the rolling window is considered in each planning step (with fixed setup decisions) to allow for adjustments of production quantities in this time interval, e.g., to avoid a setup operation within the rolling window or to avoid a capacity overload if capacity lost due to setup operations has been underestimated in preceding rolling windows. The time interval following the rolling window is considered to anticipate bottlenecks of capacity in subsequent periods.

The heuristic is governed by three parameters ($\Delta/\Psi/\Phi$) relating to the size of the rolling windows and a time limit. The first parameter Δ indicates the length of the rolling window. The third parameter Φ relates to the overlap of rolling windows of two consecutive planning steps. If two rolling windows overlap, setup decisions taken in the first planning step are revised in the second planning step.

314 Cf. Stadtler (2003) pp. 488-494 for a more detailed description of the heuristic.

Thus, integrality constraints might be relaxed for some periods at the end of the (first) rolling window. This number of relaxed periods at the end of a rolling window is indicated by the second parameter Ψ. Consequently, the following relation between parameters holds.

$$\Delta > \Phi \geq \Psi \tag{6-100}$$

Δ	Length of the rolling window
Φ	Overlap of rolling windows of two consecutive planning steps
Ψ	Number of periods with relaxed integrality constraints at the end of each (except for the last) rolling window

In the last planning step no periods at the end of the rolling window are allowed to be relaxed, because in this planning step setup decisions for the remaining periods need to be made. To identify the last planning step and to avoid that the last rolling window gets too big, the last planning step is defined to be reached, if the remaining number of periods is less than or equal to $1.5 \cdot (\Delta - \Phi)$.

The time limit is allocated to the planning steps such, that the first planning steps gets 125% of the time it would have received, if the time limit would have been equally distributed between all planning steps. The last planning step gets 75% of the time it would have received, if the time limit would have been equally distributed between all planning steps. All other planning steps get a time limit, which is linearly interpolated between these two extremes. However, there is one exception to this rule. The last planning step receives 125% instead of 75% of the time it would have received, if the time limit would have been equally distributed between all planning steps, because in this planning step the overall solution is determined.

6.2.2 Rolling Scheme

The rolling scheme is illustrated in Fig. 6-5. Like in Stadtler's (2003) approach, the planning horizon is split into three parts in each planning step: the rolling window and intervals preceding and following the rolling window. Whereas in the original approach a tight SPL reformulation of the MLCLSP has been used for the rolling window and a standard I&L model formulation for the other two intervals, here the proposed model formulations are used within the rolling window, whereas only capacity and inventory balance constraints are needed in the two other intervals. Setup decisions (binary variables) are fixed in capacity constraints corresponding to periods preceding the rolling window, while these decisions are anticipated differently in periods following the rolling window. The different kinds of anticipation used will be described in detail in the next section 6.2.3.

This means, if the heuristic is adapted for example for the CLSPL, that the extended model formulation with valid inequalities[315] is used within the rolling window. Here, binary setup decisions are made in each planning step for periods be-

[315] See section 6.1.1.2.

longing to the associated rolling window. However, production quantities are allowed to vary freely in periods preceding the current rolling window as long as capacity requirements are respected and a corresponding setup decision has been fixed in a preceding planning step.

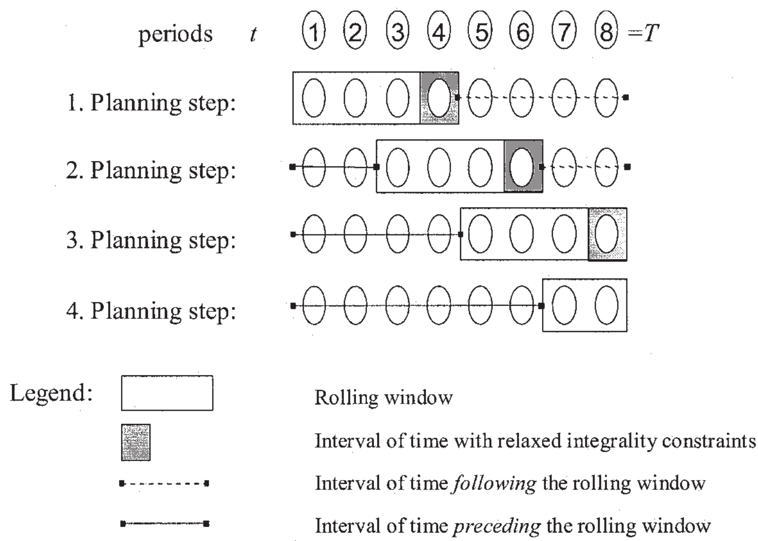


Fig. 6-5: Rolling scheme of temporal decomposition heuristic (parameter combination $\Delta/\Psi/\Phi \rightarrow 4/1/2$, slightly adapted from Stadtler (2003) p. 491)

Regarding the proposed lot size extension, this is not a valid approach. If batch size restrictions are present, production quantities have to be in integer multiples of a predefined batch size. Thus, integer variables are coupled with production quantities and therefore the following dilemma arises: If production quantities are allowed to vary freely, restrictions to obey lot size restriction have to be imposed not only inside the rolling window but also in the interval preceding the rolling window. On the other hand, if production quantities were fixed in preceding planning steps like setup decisions have been fixed, the heuristic might loose too many degrees of freedom to rearrange demands in the interval preceding the rolling window.

Therefore, two variants have been implemented, if batch size restrictions need to be obeyed. The first variant (*var*) allows production quantities to vary freely in the time interval preceding the rolling window. Consequently, constraints calculating period overlapping lot sizes and constraints obeying the associated restrictions need to be defined not only inside the rolling window, but also in the interval preceding the rolling window. This increases the size of the matrix and especially the number of integer variables in each planning step. Especially in the last planning steps model formulations with a huge number of integer variables need to be solved. Thus, this variant of the temporal decomposition heuristic is expected to need a lot more time than its original variant or the variant (*fix*) which will be explained next.

Fixing production quantities in the complete interval preceding the rolling window does not seem to be a too good idea, because it has been observed for the MLCLSP as well as for the CLSPL, that adjustments of production quantities in periods preceding the rolling window are made frequently. Preliminary tests with a model formulation respecting batch size restrictions confirmed this notion and resulted in many test instances, for which no feasible solution could be obtained. Therefore, the second variant (*fix*) does a mixture of fixing production quantities on one side and allowing additional production in the interval preceding the rolling window on the other side.

The procedure is as follows: In each planning step, the proposed model formulation (i.e. the constraints defining the lot size extension) is only defined inside the rolling window. Setup decisions and production quantities are fixed in each planning step for those periods preceding the rolling window. On the other hand, to allow for some flexibility in periods preceding the rolling window, inventory balance constraints are amended such that for each product a certain number of *a*dditional *b*atches (AB_j) might be "bought" at the end of the period preceding the rolling window (T^{fix}) (6-101). The number of batches, that is allowed to be bought ($addbat_j$), is limited to the minimum of three and $\lceil T^{fix}/J \rceil$ (6-102).

Furthermore, the number of batches to be bought is limited by the number of batches that can be additionally produced in the interval preceding the rolling window. To obtain this number, the capacity not used so far is calculated for each period (fc_t), and it is evaluated whether the lots already planned might be increased by additional batches without violating capacity constraints or lot size restrictions. E.g., if a lot starts in period 13 and continues to be produced in period 14, it is evaluated, whether the remaining capacity in periods 13 and 14 allow for production of one (or more) additional batch(es) of this product. Fig. 6-6 describes the algorithm to derive the number of batches that are allowed to be bought ($addbat_j$) and their associated costs ($addcost_j$) in detail.

$$I_{jt-1} + x_{jt}^{fix} + bs_j \cdot AB_j = d_{jt} + I_{jt} \qquad \forall j \in \mathcal{J},\ t = T^{fix} \qquad (6\text{-}101)$$

$$AB_j \le addbat_j \qquad \forall j \in \mathcal{J} \qquad (6\text{-}102)$$

Data:

$addbat_j$	Maximal number of batches of product j allowed to be bought in T^{fix}
$addcost_j$	Additional cost incurred if one batch of product j is bought in T^{fix}
fc_t	Free capacity in period t based on fixed setup and production decisions
fpc_{jt}	Free capacity in period t to produce product j based on fixed setup and production decisions
T^{fix}	Number of periods preceding the rolling window (= number of periods with fixed setup decision)
x_{jt}^{fix}	Fixed production quantity of product j in period t

Variables:

AB_j Integer number of additional batches of product j bought in T^{fix}

- $t = 1..T^{fix}$
  - Compute free (unused) capacity in period t (fc_t)
  - $j = 1..J$
    - IF product j is producible in period t based on the (fixed) setup pattern
      - THEN: Compute free capacity of period t to produce product j ($fpc_{jt} = fc_t$)
      - ELSE: Compute free capacity of period t to produce product j ($fpc_{jt} = 0$)
- Initialize maximal number of additional batches allowed for each product j ($addbat_j = 0$)
- $t = 1..T^{fix}$
  - $j = 1..J$
    - $i = 1..3$
      - IF one batch (bs_j) of product j is producible in period $T^{fix}+1-t$ based on
        - the free capacity of period $T^{fix}+1-t$ to produce product j ($fpc_{j,Tfix+1-t}$) and
        - the maximum lot size restriction of the current lot ($maxlot_j$)

        AND $addbat_j < \min\{3; \lceil T^{fix}/J \rceil\}$
        - THEN:
          - $addbat_j = addbat_j + 1$
          - $addcost_j = [(t-1) \cdot h_{jt} \cdot bs_j + (addbat_j - 1) \cdot addcost_j] / addbat_j$
          - $k = 1..J$
            - $fpc_{k,Tfix+1-t} = fpc_{k,Tfix+1-t} - a_j \cdot bs_j$
        - ELSE:
          - IF one batch (bs_j) of product j is producible in period $T^{fix}+1-t$ and $T^{fix}-t$ combined based on
            - the (fixed) setup pattern and
            - the free capacity of period $T^{fix}+1-t$ and $T^{fix}-t$ ($fpc_{j,Tfix+1-t} + fpc_{j,Tfix-t}$) and
            - the maximum lot size restriction of the current lot ($maxlot_j$)

            AND $addbat_j < \min\{3; \lceil T^{fix}/J \rceil\}$
            - THEN:
              - $addbat_j = addbat_j + 1$
              - $addcost_j = [(t-½) \cdot h_{jt} \cdot bs_j + (addbat_j-1) \cdot addcost_j] / addbat_j$
              - $k = 1..J$ ($k \neq j$)
                - $fpc_{k,Tfix+1-t} = fpc_{k,Tfix+1-t} - fpc_{j,Tfix+1-t}$
                - $fpc_{k,Tfix-t} = fpc_{k,Tfix-t} - a_j \cdot bs_j + fpc_{j,Tfix+1-t}$
              - $fpc_{j,Tfix-t} = fpc_{j,Tfix-t} - a_j \cdot bs_j + fpc_{j,Tfix+1-t}$
              - $fpc_{j,Tfix+1-t} = 0$
            - ELSE

Fig. 6-6: Algorithm to derive $addbat_j$ and $addcost_j$ for variant *fix*.

Finally, a cost has to be associated with this purchase of additional batches ($addcost_j$). To keep it simple, the cost is calculated to match the mean holding cost, which would be incurred if all of the batches allowed to be bought are taken. E.g., if the evaluation of capacity constraints and lot size restrictions gives the result that one additional batch can be bought two periods in front of the rolling window and another one can be bought six periods in front of the rolling window, the cost of purchasing one batch would equal holding one batch four periods in inventory ($[6+2]/2 \cdot h_j \cdot bs_j$). The exact procedure to derive $addcost_j$ is described in Fig. 6-6. Prior to the next planning step, the additional batches (AB_j) are converted into fixed production quantities .

This variant (*fix*) requires only some preprocessing of data. The model formulation itself is only slightly altered

- in the objective function (price and number of batches "bought", $+\sum_{j \in J} addcost_j \cdot AB_j$),
- in inventory balance constraints of period T^{fix} (purchase of additional batches, (6-101)),

and supplemented with J upper bound constraints limiting the purchase of additional batches (6-102). Thus, the additional computational effort associated with this variant to model the batch size extension is negligible compared to variant (*var*).

However, variant (*fix*) allows only to schedule additional batches, if enough capacity for one batch is left over in two consecutive periods combined. This means, already fixed lots are not allowed to be moved to obtain space for an additional batch. In tightly capacitated test instances, the following situation may occur (see also Fig. 6-7). Due to fixed lot sizes, in each period 40 units of capacity remain unused. One batch requires 100 units of capacity. Thus, there are no two consecutive periods available to schedule one additional batch (of product 2 in Fig. 6-7). On the other hand, by shifting the preceding lot 20 units of capacity back in time (product 1), the additional batch could have been scheduled.

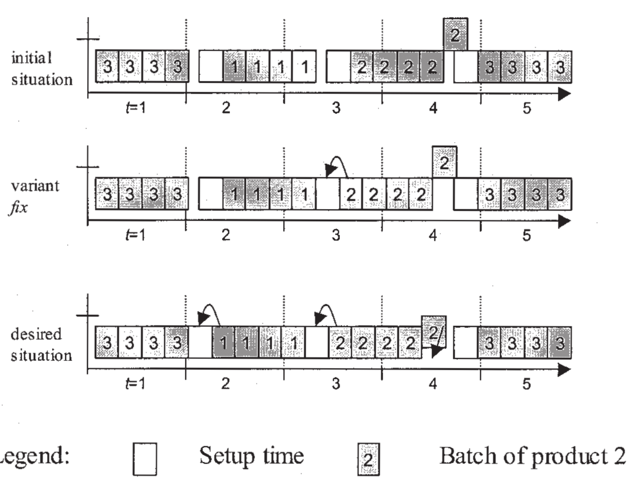


Fig. 6-7: Illustration of the desired lot movement for variants *fix* and *fix1*.

Therefore, a small adaptation of variant *fix* called *fix1* will be proposed. This variant does not consider all possible movements a priori. Instead, it allows the purchase of just one additional batch of each product in each planning step, in which the interval preceding the rolling window is greater than five periods. Thereby, it is assumed that after more than five periods enough unused capacity has been accumulated to make such a movement of lots possible. In the whole planning interval, only one such additional purchase per product is allowed at a very high cost ($>>sc_j$, $>>h_j$). However, these additional purchases are not converted into fixed production quantities x_{jt}^{fix} in each planning step. Only after the last planning step has been carried out, these additional batches are to be scheduled. Therefore, the complete model formulation is solved again with fixed setup decisions after the last planning step. Although there might be situations, in which such a movement of lots will not be possible, the computational tests revealed that these situations are rather seldom.

6.2.3 Anticipation

From a computational point of view it is desirable to keep the rolling windows as small as possible, because this reduces the number of binary/integer variables in each planning step. On the other hand, short rolling windows suffer the most from planning horizon effects.[316] To alleviate these effects as well as to anticipate capacity bottlenecks in periods following the rolling window, several measures will be taken.

The interval following each rolling window is considered in each planning step to anticipate future capacity bottlenecks. As long as no setup times have to be respected, this anticipation is fine. However, if setup times occur, these have to be anticipated explicitly. This anticipation is important, because if future capacity losses due to setup times are overestimated, this may lead to too many setup operations in early periods, while an underestimation might result in infeasible solutions. For the MLCLSP, Stadtler (2003) proposes three options.[317] The first one is to neglect setup operations in periods following the lot-sizing window, while the second option is to assume that each product is produced each period. Hence, the first option will underestimate setup times and the second option will overestimate setup times. The third option therefore assumes that a mean setup time plus some safety margin will occur in each period. Alongside these rather simple measures to anticipate setup times, also more accurate measures, e.g. by means of neural networks, have been proposed.[318] However, we will stick to these simpler measures here.

The first option "*min*" assumes that no capacity is lost due to setup operations in the time interval preceding the rolling window. If the CLSPL is taken as the ba-

316 See section 5.2.2.

317 Cf. Stadtler (2003) p. 494.

318 E.g., Rohde (2004) anticipates capacities lost due to setup times in a PLSP with a neural network approach.

sic model, the second option "*max*" assumes that each product except the one with the smallest setup time is set up each period. On the other hand, if the PLSP is taken as the basic model, this option assumes that the product with the biggest setup time is set up each period. Anyhow, with this option, many tightly capacitated test instances have resulted in infeasible solutions of the first planning step.[319] Thus, this option has not been tested much further.

In the third option "*mean*" it is assumed that in each period of the time interval following the rolling window the same amount of setup time is incurred like it has been incurred on the average in periods of the interval preceding the rolling window. As the mean is not meaningful in the first periods, it is only used, if the number of periods in the time interval preceding the rolling window is greater than four. In prior planning steps the expected capacity lost due to setup operations is calculated to be $\sum_{j\in\mathcal{J}} st_j / tbo_j$. This means that it is assumed that each product j is on average set up according to its expected time between orders tbo_j. The expected time between orders tbo_j is calculated assuming production lots according to the EOQ-formula $tbo_j = \sqrt{2 \cdot sc_j \cdot T / (h_j \cdot \sum_{t\in\mathcal{T}} d_{jt})}$.[320]

If the PLSP is used as a basic model, an anticipation of setup times may not suffice to obtain feasible schedules in each planning step. One can think of a situation, in which three (or more) products face positive demands in the first period of a rolling window, but at most two products can be produced in this period due to the setup restriction of the PLSP. Therefore, it seems wise not to relax integrality constraints in (some of) the overlapping periods to better anticipate future setup patterns.

Finally, to overcome the planning horizon effect Stadtler (2000, 2003) proposed the use of bonuses.[321] The idea is to make setup decisions at the end of the rolling window more attractive. Thus, not a full setup cost is charged for a setup operation in the last period(s) of the rolling window, but only a portion of the setup cost. Prior to each planning step, for each period and product a myopic time between orders is calculated with the Silver-Meal or Groff heuristic. Then setup cost coefficients are updated such that only the fraction of periods inside the rolling window by this myopic time between orders of the full setup cost is used as a cost coefficient for the specific period. E.g., if the myopic time between orders was four periods for the last period of the rolling window, the updated setup cost coefficient would be only one quarter of its original value. Furthermore, bonuses are not only granted for setup costs, but also for inventory costs associated with these setup decisions.

Although this calculation is exact for the single-level uncapacitated lot-sizing problem, it is only an approximation for the MLCLSP.[322] Within the setting evaluated here, which allows to carry-over setup states, it is even a worse approximation. The reason for this is that bonuses are intended to reduce setup operations by not imposing a setup pattern dependent on the size of the rolling window. If setup

---

319 See Table 7-19 in section 7.1.3.

320 Cf. Salomon (1991) p. 115.

321 Cf. Stadtler (2000) pp. 320-321 and Stadtler (2003) pp. 491-492, 494.

322 Cf. Stadtler (2003) p. 492.

carry-over is allowed, products with higher setup costs are associated with the highest bonuses, but exactly these are the products, which are rather rarely set up, but more often carried-over. Still, incorporation of bonuses proved to be a viable approach for the CLSPL and has been used again for this model here (option: *bonus*).[323]

[323] Cf. Suerie and Stadtler (2003) p. 1048.

7 Analysis of Solutions and Computational Performance

In this chapter the performance of the proposed model formulations and the temporal decomposition heuristic will be reviewed. This review is based on two pillars. One pillar is the analysis of solutions. This part is intended to give insights, in what makes specific situations difficult to solve, and thereby provides managerial insights. The second pillar is the assessment of computational performance. In this part, the solution approach is compared to benchmark approaches if suitable benchmarks exist in literature. If no suitable benchmark can be obtained, the computational performance is analyzed with respect to different problem sizes and different variants of the solution approach.

This chapter is organized similar to chapters 3 and 6.1. Sections 7.1–7.4 review the different aspects of time continuity introduced (setup states, lot sizes, setup operations and resource utilization). In section 7.5 further extensions are discussed and section 7.6 compares the influence of solver technology used on the proposed model formulations.

In each section, the test set(s) and benchmark(s) are introduced before solutions are analyzed and computational performance is assessed.

7.1 Time Continuity – Setup States

The focus in this section will be on the basic model CLSPL, because a new model formulation has been proposed to solve it in section 6.1.1.2. Furthermore, it is the more difficult problem, as it allows more degrees of freedom than the PLSP.

7.1.1 Test Sets and Benchmarks

The CLSPL has been tackled by various authors before.[324] Three approaches will serve as a benchmark here. The MIP model formulations by Gopalakrishnan (2000)[325] and Sox and Gao (1999)[326] have been implemented, so that they can run

---

324 See sections 2.4.1 and 4.1.3.

325 Cf. Gopalakrishnan (2000) pp. 3421-3423. The model formulation can be found in the appendix.

326 Cf. Sox and Gao (1999) pp. 175-176. The model formulation of Sox and Gao (1999) does not consider setup times, which are present in the test set used here, and contains

on the same hardware and software for a fair comparison of model formulations. Furthermore, the tabu search algorithm by Gopalakrishnan et al. (2001)[327] is a suitable benchmark. This algorithm has not been re-implemented, but the authors used one of the test sets (*TT*) which has also been used here.

The first test set *TT* was developed by Trigeiro et al. (1989) for the CLSP,[328] but has been used subsequently also for the CLSPL.[329] Test set *TT* is divided into three phases. For the first phase no results are reported in literature, because this phase is used for parameter tuning of the algorithm etc. The test instances from phases II and III can be clustered into nine classes according to the number of products and the number of time periods (see Table 7-1). The classes will be referenced as test sets *TT*1–*TT*9 from this point.

Table 7-1: Test set *TT*.

Phase	II	II	II	II	II	II	III	III	III
Class	1	2	3	4	5	6	7	8	9
# products	6	6	12	12	24	24	10	20	30
# periods	15	30	15	30	15	30	20	20	20
# instances	116	5	5	5	5	5	180	180	180

Although test set *TT* has been used in literature to assess computational performance of the CLSPL, it does not seem to be an appropriate test set for model CLSPL in its entirety. If up to 30 products are produced in each period (test set *TT*9), there is not much to gain, if only one setup state is carried over from one period to the next. Furthermore, it is highly unlikely, that there exist periods with production of only one item (single-product production), which is an important feature modeled by the CLSPL.

The CLSPL is intended for use in situations in which setup operations are substantial. This means, there has to be a considerable amount of cost or use of capacity associated with a setup operation. Only in these situations it makes sense to model the preservation of setup states explicitly. At the same time, this means that usually only few items will be produced in each period.

Taking these reflections into account, three new test sets *TL*1–*TL*3 are derived based on test sets *TT*7–*TT*9. Test set *TL*1 originates from test set *TT*7 by aggregating products 1–4 and 5–8 to form two new products. Together with products 9 and 10 of the original test set *TT*7 these two aggregated products form test set *TL*1, which now consists of four products. The aggregation has been defined such, that the sum of demands (setup times, setup costs) is taken as the demand (setup time, setup cost) of the new product, whereas the average is taken for holding cost and

a bug. Therefore, the model formulation has been adapted according to Suerie and Stadtler (2003) pp. 1053-1054 to overcome these two deficiencies. The (corrected) model formulation can also be found in the appendix.

327 Cf. Gopalakrishnan et al. (2001) pp. 853-859.

328 Cf. Trigeiro et al. (1989) pp. 358-363.

329 Cf. Gopalakrishnan et al. (2001) pp. 860-862.

production coefficients. This kind of aggregation could well be the result of an aggregation of items to product families.

Test set *TL*2 is derived from test set *TT*8 by aggregating products 1–8 and 9–16. Consequently, test set *TL*2 consists of six products, two of them being an aggregate. Finally, test set *TL*3 is derived from test set *TT*9 by aggregating products 1–10, 11–20, 21–23 and 24–26. It consists of eight products (four of them being an aggregate).

7.1.2 Analysis of Solutions

Difference of CLSP and CLSPL Solutions

Several authors have analyzed the difference of optimal solutions of models that allow for setup preservation at period boundaries (e.g., CLSPL) and those that do not (e.g., CLSP) and claimed to have found fundamental deviations.[330] These fundamental deviations justify the additional computational effort associated with this modeling option, yielding a more natural representation of the shop floor in the model. Several observations derived from the computational analysis conducted here support this claim.

Table 7-2 shows the optimal setup pattern of the first test instance of test set *TT*1 of model formulations CLSP and CLSPL. The setup pattern of the optimal solution to the CLSP is marked in gray, whereas setup operations (setup carry-overs) in the optimal solution of model CLSPL are identified by an X (O). Although the set of products that are produced in the first two periods is the same for both solutions, it is different in every single period thereafter. Thus, as soon as model CLSPL is able to generate enough "additional" capacity by the more exact modeling of setup times, this changes the setup pattern in subsequent periods.

Table 7-2: Comparison of setup pattern for the first test instance of test set *TT*1 of optimal solutions to CLSP and CLSPL.

$j \backslash t$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	X	O		X	O		X	O		X	O		X	O	
2	X	X		X		X	O		X	O		X		X	O
3	X		X		X		X		X		X			X	
4	X	X	O		X	O		X	O			X	O		
5			X	X		X		X		X		X		X	X
6	X		X	O	X		X	X		X	X	O	X		X

Legend:
- (gray) Setup operation in optimal solution of CLSP
- X Setup operation in optimal solution of CLSPL
- O Setup carry-over in optimal solution of CLSPL

In Table 7-3 the objective function values (total costs) and its components (inventory holding costs and setup costs) are compared for optimal or near optimal

[330] Cf. Haase (1998) pp. 140-143, Sox and Gao (1999) pp. 176-178 and Porkka et al. (2003) pp. 1141-1146.

solutions of the CLSP and CLSPL of test set *TT*1.[331] It is interesting to note, that although solutions to the CLSPL are on average 20.58% less expensive than solutions to the CLSP, the ratio of inventory holding costs to setup costs stays more or less the same (30:70). Consequently, in absolute values the setup costs are decreased farther than inventory holding costs (13.35% vs. 7.23%). There exist even some test instances, in which inventory holding costs are increased, which is of course offset by a massive decrease in setup costs.

Table 7-3: Comparison of optimal (or near optimal) costs in models CLSP and CLSPL for test set *TT*1.

	CLSP	CLSPL
Mean percentage of inventory holding costs	30.13%	28.79%
- standard deviation	4.99%	4.27%
Mean percentage of setup costs	69.87%	71.21%
- standard deviation	4.99%	4.27%
Mean cost saving vs. CLSP	-	20.58%
- minimum	-	15.76%
- maximum	-	26.34%
Mean savings (inventory costs) vs. CLSP	-	7.23%
- minimum	-	-2.72%
- maximum	-	18.13%
Mean savings (setup costs) vs. CLSP	-	13.35%
- minimum	-	2.63%
- maximum	-	25.09%

Finally, Table 7-4 analyzes the setup patterns for the same set of solutions. As all products in test set *TT*1 have similar characteristics, no differences are observed regarding the different products. Comparing solutions of the CLSP and the CLSPL, it becomes obvious that due to the setup carry-over feature the number of periods a specific product can be produced increases from 7.7 for the CLSP to 8.9 for the CLSPL despite a decrease in setup operations to 6.5. Thereby, the flexibility to schedule products in certain periods increases (which allows for a decrease in inventory holding costs), while the cost of the schedule decreases (in terms of setup costs).

In Table 7-5 CLSPL solutions are analyzed.[332] Here, the conjecture is confirmed that test sets *TT*1–*TT*9 are not as appropriate as test sets *TL*1–*TL*3 to assess

331 Model CLSP (see section 2.2) and the extended formulation in SPL reformulation with valid inequalities of model CLSPL (see sections 5.2.1.1 and 6.1.1.2) have been given a time limit of 15 minutes to solve each of the 116 test instances of test set *T*1. All but 28 (17) test instances have been solved to proven optimality with model CLSP (CLSPL). See Table 7-9 for detailed solution statistics.

332 The extended formulation in SPL reformulation with valid inequalities of model CLSPL (see sections 5.2.1.1 and 6.1.1.2) with time limits of 900s (test set *T*1) and 60s (all other test sets) has been used to obtain the solutions. If no solution has been found within the time limit the first solution is taken instead. See Table 7-13–Table 7-16 for solution statistics.

the behavior of the CLSPL. Its feature to produce only one single product for a complete period (last column) is used either rarely (in test sets *TT*1, *TT*7, *TT*8 and *TT*9) or not at all (test sets *TT*2–*TT*6). In contrast, it is used massively in 20.5% of all periods in test set *TL*1 and somewhat less in test sets *TL*2 and *TL*3 due to the fact, that more products have to be scheduled in these test sets.

Table 7-4: Number of setups in optimal (or near optimal) solutions to models CLSP and CLSPL for test set *TT*1.

	CLSP			CLSPL		
	mean	min	max	mean	min	max
Setup operations for *j*=1	7.7	3	15	6.5	3	15
Setup operations for *j*=2	7.7	4	15	6.7	3	15
Setup operations for *j*=3	7.6	3	15	6.5	3	15
Setup operations for *j*=4	8.3	4	15	6.9	3	15
Setup operations for *j*=5	7.5	3	15	6.4	3	15
Setup operations for *j*=6	7.6	2	14	6.2	2	14
Setup links for *j*=1	-	-	-	2.3	0	5
Setup links for *j*=2	-	-	-	2.1	0	6
Setup links for *j*=3	-	-	-	2.0	0	5
Setup links for *j*=4	-	-	-	2.6	0	6
Setup links for *j*=5	-	-	-	2.4	0	7
Setup links for *j*=6	-	-	-	2.6	0	6

Table 7-5: Number of setup operations, setup carry-overs (per product and test instance) and single-product production periods (per test instance) in solutions to model CLSPL for different test sets.

Test	Setup operations			Setup Carry-overs			Single-product
set	mean	min	max	mean	min	max	period
*TT*1	6.5	2	15	2.3	0	7	0.2
*TT*2	14.9	7	28	4.8	0	10	-
*TT*3	7.6	3	15	1.2	0	5	-
*TT*4	14.2	6	30	2.4	0	9	-
*TT*5	7.5	3	15	0.6	0	4	-
*TT*6	14.7	6	30	1.2	0	9	-
*TT*7	9.0	3	20	1.9	0	9	0.3
*TT*8	9.4	2	20	1.0	0	9	0.1
*TT*9	9.5	3	20	0.6	0	9	0.0
*TL*1	8.0	2	20	4.7	0	12	4.1
*TL*2	8.6	2	20	3.1	0	10	3.8
*TL*3	9.0	3	20	2.4	0	10	2.5

Due to the aggregation procedure, the products in test sets *TL*1–*TL*3 have different characteristics opposed to those in test sets *TT*1–*TT*9. Remember, that several products of test sets *TT*7–*TT*9 have been aggregated to form product families in test sets *TL*1–*TL*3.[333] Thereby, these test instances contain products with differ-

333 See section 7.1.1.

ent characteristics. E.g., test set *TL*1 is made up of four products, with two of them facing a fourfold demand as the other two. Consequently, the analysis in Table 7-5 needs to be product group dependent for test sets *TL*1–*TL*3 to reveal additional insights. Table 7-6 provides this analysis. As expected, products with higher demand are set up less frequently, but their setup state is carried over much more often. Thereby, these products are produced on average in more periods than those products with a lower demand profile. Furthermore, products with a higher demand rate are far more often produced for complete periods (single-product production periods), which obviously makes sense.

Table 7-6: Number of setup operations, setup carry-overs and single-product production periods (per product and test instance) in solutions to model CLSPL for test sets *TL*1–*TL*3 depending on product group (high, medium and low demand profile).

Test set	Demand profile	Setup operations mean	min	max	Setup Carry-overs mean	min	max	Single-product period mean	min	max
*TL*1	high	6.7	3	11	8.2	5	12	2.0	0	7
	low	9.2	2	20	1.3	0	6	0.1	0	1
*TL*2	high	7.0	3	11	8.2	5	10	1.9	0	6
	low	9.4	2	20	0.6	0	4	0.0	0	1
*TL*3	high	7.0	3	11	7.5	3	10	1.2	0	6
	medium	9.8	3	20	1.6	0	6	0.0	0	1
	low	9.6	3	20	0.2	0	3	0.0	0	1

Effectiveness of Valid Inequalities

Table 7-7 and Table 7-8 intend to demonstrate the effect of using an extended formulation and the addition of valid inequalities. In Table 7-7 the solution of the linear relaxation[334] of the basic model CLSPL[335] is shown for the first test instance of test set *TT*1. In contrast, Table 7-8 shows the solution of the linear relaxation for the same test instance, if the extended formulation in SPL reformulation with valid inequalities of model CLSPL[336] is taken.

It is obvious, that the solution to the linear relaxation of the extended model formulation with valid inequalities contains a lot less fractional values for the binary variables Y_{jt} and W_{jt} than the solution to the linear relaxation of the basic model formulation. Thereby, the number of variables which are potentially used for branching is considerably reduced (here, in the root node, from 106 to 36). Furthermore, a lot more variables take the value “1” representing a setup operation or setup carry-over in the non-basic model formulation (here: 39 vs. 7 in the basic model formulation). This means, that nearly no (positive) decision is fixed in the basic model formulation. In addition, the number of fractional values that are greater than 0.5 hinting to positive decisions is bigger in the non-basic model for-

334 In fact, not the linear relaxations of the original model formulations are compared, but the linear relaxations after presolving of the MIP solver (here: XpressMP, release 2003G), which itself improves the linear relaxation considerably.

335 See section 2.4.1.

336 See sections 5.2.1.1 and 6.1.1.2.

mulation (17 vs. 4 in the basic model formulation). As 55 binary variables take the value "1" in the optimal solution, this is much more reflected in the non-basic model formulation than in the basic model formulation.

Going into more detail, especially the value "1" of link variables W_{jt} is (almost evenly) split over all products in the linear relaxation of the solution to the basic model formulation. Taking the view of the linear relaxation, this makes a lot of sense, since a positive (fractional) value of variables W_{jt} allows production (2-32), but does not cost anything, as no setup costs are associated with this variable. On the other hand, to obtain a feasible solution, fractional values for any binary variables need to be avoided.

Table 7-7: Linear relaxation for the first test instance of test set *TT*1 of basic model formulation CLSPL.

$j \backslash t$	Variable	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Y_{jt}		0.10		0.01		0.01		0.01		0.04	0.02	0.06	0.08	0.34	
	W_{jt}			0.10	0.10	0.11	0.11	0.12	0.12	0.12	0.13	0.16	0.19	0.25	0.33	0.67
2	Y_{jt}	1.00								0.03	0.02	0.14	0.24	0.29	0.49	1.00
	W_{jt}		0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.15	0.05		0.02		
3	Y_{jt}	1.00										0.01	0.03	0.07	0.87	
	W_{jt}		0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.21	0.24	0.13	
4	Y_{jt}	1.00										0.02	0.04	0.14	0.50	1.00
	W_{jt}		0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.20	0.24		
5	Y_{jt}	1.00												0.29		
	W_{jt}		0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.54	0.33
6	Y_{jt}		0.10					0.01	0.01	0.02	0.07	0.03	0.08	0.34	0.52	1.00
	W_{jt}			0.10	0.11	0.11	0.11	0.11	0.11	0.12	0.09	0.16	0.16			

Legend:

Y_{jt}	Linear relaxation of setup operation variable Y_{jt}
W_{jt}	Linear relaxation of link variable W_{jt}

Table 7-8: Linear relaxation for the first test instance of test set *TT*1 of the extended formulation in SPL reformulation with valid inequalities of model CLSPL.

$j \backslash t$	Variable	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Y_{jt}		1.00				1.00			0.24	0.76			1.00		
	W_{jt}			1.00	1.00			1.00	0.76		0.24	1.00			1.00	
2	Y_{jt}	1.00			1.00		1.00	0.24	0.76	1.00	0.76	0.24	0.76	0.24	1.00	1.00
	W_{jt}															
3	Y_{jt}	1.00				1.00		0.24	0.60	0.16	0.24	0.76	0.15	0.09	0.91	
	W_{jt}						0.16						0.61			
4	Y_{jt}	1.00			1.00	1.00	0.16		1.00	0.76		1.00		1.00	1.00	
	W_{jt}						0.84			0.24	0.76		0.15			1.00
5	Y_{jt}	1.00			1.00			0.24	0.76			0.24	0.76			
	W_{jt}		1.00			1.00			0.24	0.76			0.24	1.00		
6	Y_{jt}		1.00		1.00	1.00	1.00	0.24	1.00	1.00	0.76	1.00	0.76	1.00	1.00	1.00
	W_{jt}															

Legend:

Y_{jt}	Linear relaxation of setup operation variable Y_{jt}
W_{jt}	Linear relaxation of link variable W_{jt}

Reviewing this analysis, it proved worthwhile to enhance the modeling capabilities of the standard lot sizing problem CLSP by adding the functionality to carry over setup states across period boundaries. Solutions to the CLSP are fundamentally different to those of the CLSPL. Especially, if the number of products to be scheduled is small, the modeling capabilities of the CLSPL are useful. Then, even the feature to produce one product for three (or more) consecutive periods, i.e. to have at least one period with production of only one product, is used rather frequently.

7.1.3 Computational Performance

All computational results in this section have been obtained using XpressMP release 2003G as a MIP solver on a PC equipped with a Pentium IV processor with a clockspeed of 1.7 GHz and 256 MB of memory.

Table 7-9 compares the computational performance of the basic model CLSP[337] and the extended formulation in SPL reformulation with valid inequalities of model CLSPL[338]. Both have been given a time limit of 15 minutes to solve each of the 116 test instances of test set *TT*1. All but 28 (17) test instances have been solved to proven optimality with model CLSP (CLSPL). The remaining optimality gap[339] between the best solution obtained within 15 minutes and the best bound obtained after 15 minutes has been on the average 4.54% (2.27%) for the 28 (17) test instances not solved to proven optimality within the time limit for the CLSP (CLSPL). The average time needed to prove optimality was 114.5s for the CLSP and 107.0s for the CLSPL. Thus, the best model formulation of the CLSPL compares favorably even to a basic model formulation of the CLSP.

Table 7-9: Comparison of CLSP and CLSPL for test set *TT*1.

	CLSP	CLSPL
Average time to prove optimality (if less than 900s) [s]	114.5	107.0
# not optimal after 900s	28	17
Remaining gap after 900s (for those not optimal)	4.54%	2.27%
Average cost saving vs. CLSP	–	20.65%

To analyze the behavior of different model formulations in more detail, the following notation is introduced to distinguish between these model formulations (Table 7-10). The first field represents the basic model formulation, which is either based on the standard inventory and lot size ("I&L") format or on the simple

337 See section 2.2.

338 See sections 5.2.1.1 and 6.1.1.2.

339 Different "gaps" are calculated to assess computational performance in this thesis. The gaps are always defined as the percentage deviation of an upper bound (UB) to a lower bound (LB) [=(UB–LB)/LB]. Here the upper bound is the best solution obtained within the time limit of 15 minutes and the lower bound is the bound obtained by the branch-and-bound search after 15 minutes.

plant location ("SPL") formulation. The second field represents the different variants to model the "linking" of lots in the CLSPL. Possible values for this field are "basis", "ext" and "uss" as described in Table 7-10. Finally, the third field describes the use of valid inequalities. These are either used ("vi") or not ("–"). If valid inequalities are used, they can be applied at the root node only (cut-and-branch approach, "C&B") or throughout the branch-and-bound search. The latter is indicated by "B&C". Here, the model cut feature of XpressMP is used. E.g., the abbreviation "I&L/uss/vi" describes a model formulation with inventory and lot size variables ("I&L"), in which the setup state is uniquely defined ("uss") and valid inequalities ("vi") have been added to the model formulation.

Table 7-10: Notation of different model formulations (CLSPL).

Parameter	Value	Comment
1	I&L	Standard basic model formulation with variables for inventories and lot sizes (*I&L*)
	SPL	Simple plant location (*SPL*) reformulation of the basic model formulation (see section 5.2.1.1)
2	basis	Basic model formulation (see section 2.4.1)
	ext	*Ext*ended model formulation (see section 6.1.1.2)
	uss	Model formulation with *u*nique *s*etup *s*tates (see section 6.1.1.2)
3	–	No valid inequalities
	vi	Valid inequalities (see section 6.1.1.2)
	vi-C&B	Valid inequalities (see section 6.1.1.2), *C*ut&*B*ranch approach (=valid inequalities only applied in the root node, see section 5.2.1.2)
	vi-B&C	Valid inequalities (see section 6.1.1.2), *B*ranch&*C*ut approach (=model cut feature of XpressMP, see section 5.2.1.4)

Test Set *TT*1

In Table 7-11 and Table 7-12 the capabilities of different model formulations are analyzed based on test set *TT*1. In Table 7-11 the best solution found within 20 seconds is used as an upper bound which is set in relation to various lower bounds. If no solution could be obtained within this time limit (number in parenthesis) the first solution is taken instead. This solution is found after a maximum of 21s for I&L/ext/vi-C&B and 39s for Gopalakrishnan. The different lower bounds are the linear relaxation (LB^{LP}), the lower bound obtained after cuts have been generated automatically by XpressMP (LB^{XLP}), the lower bound obtained after cuts have been generated automatically by XpressMP for model SPL/ext/vi-C&B ($LB^{XLP(SPL/ext/vi)}$) and the optimal solution (LB^{OPT}). Model formulations I&L/basis/–, Gopalakrishnan as well as Sox and Gao are intended as a benchmark.

One can observe, that the best model formulations are those that use the SPL reformulation as a basic model, the extended formulation to model the "linking" feature and valid inequalities. They provide the best lower bounds (see columns 2 and 3 in Table 7-11) as well as the best solutions (see columns 4 and 5 in Table 7-11). Both produce significantly better solutions than the three model formulations used as benchmarks based on the one-sided Wilcoxon matched-pairs signed-ranks test, which has been used here at confidence limits of 99%. Furthermore, all model formulations that make use of valid inequalities provide on average better solutions than the three benchmark model formulations.

Table 7-11: Average gaps of solution after 20s (to different lower bounds) of different model formulations for test set *TT*1.

	Average Gap to LB (with different bases)			
Model formulation	LB^{LP}	LB^{XLP}	$LB^{XLP(SPL/ext/vi)}$	LB^{OPT}
I&L/basis/–	323.08%	21.64%	5.34%	3.49%
I&L/ext/–	320.26%	19.51%	5.62%	3.76%
I&L/ext/vi-C&B	3.77% (1)	3.09% (1)	2.99% (1)	1.19% (1)
I&L/ext/vi-B&C	283.95%	2.91%	2.49%	0.69%
I&L/uss/–	324.49%	22.31%	5.72%	2.61%
I&L/uss/vi	6.48%	5.02%	3.10%	1.29%
SPL/ext/–	13.57%	10.13%	3.88%	2.06%
SPL/ext/vi-C&B	3.73%	3.17%	3.17%	1.35%
SPL/ext/vi-B&C	12.40%	2.85%	2.62%	0.82%
SPL/uss/–	20.27%	12.72%	3.95%	2.12%
SPL/uss/vi	7.12%	5.41%	3.94%	2.10%
Sox and Gao	7.32%	6.56%	4.12%	2.29%
Gopalakrishnan	347.86% (5)	31.45% (5)	14.24% (5)	12.26% (5)

Table 7-12 depicts the worst case (maximum gap) for each model formulation. With one exception (one test instance, SPL/ext/vi-C&B) the worst case performance is much better for model formulations based on the SPL formulation than for those based on the I&L formulation. On the other hand, feasible solutions are obtained much faster, if the I&L formulation is used as a basis.

Table 7-12: Maximum gaps of solution after 20s (to different lower bounds) and solution times of different model formulations for test set *TT*1.

	Maximum Gap to LB		Avg. time 1st	Max time 1st
Model formulation	LB^{XLP}	LB^{OPT}	solution [s]	solution [s]
I&L/basis/–	45.79%	10.51%	2.2	5
I&L/ext/–	49.44%	10.84%	3.5	8
I&L/ext/vi-C&B	16.00% (1)	9.67% (1)	7.4	21
I&L/ext/vi-B&C	13.28%	6.31%	4.3	11
I&L/uss/–	50.30%	10.71%	2.6	7
I&L/uss/vi	16.70%	6.98%	6.1	14
SPL/ext/–	31.22%	7.88%	5.9	13
SPL/ext/vi-C&B	21.38%	16.06%	8.2	19
SPL/ext/vi-B&C	11.91%	4.54%	5.0	12
SPL/uss/–	30.85%	6.11%	3.9	9
SPL/uss/vi	16.55%	9.64%	8.6	17
Sox and Gao	22.04%	12.53%	7.2	20
Gopalakrishnan	68.53% (5)	36.22% (5)	10.4	39

Fig. 7-1 shows the results graphically. The solution quality after 20s compared to the optimal solutions is depicted (left scale) together with solution speed based on the average time to find a first feasible solution (right scale). Model formulations I&L/ext/vi-B&C and SPL/ext/vi-B&C are identified as the variants with the best solution quality and reasonable solution speed. On the other hand, it is obvious that among those three model formulations that find a first feasible solution the quickest (I&L/basis/–, I&L/ext/– and I&L/uss/–) are also those that have found the worst solutions within the time limit.

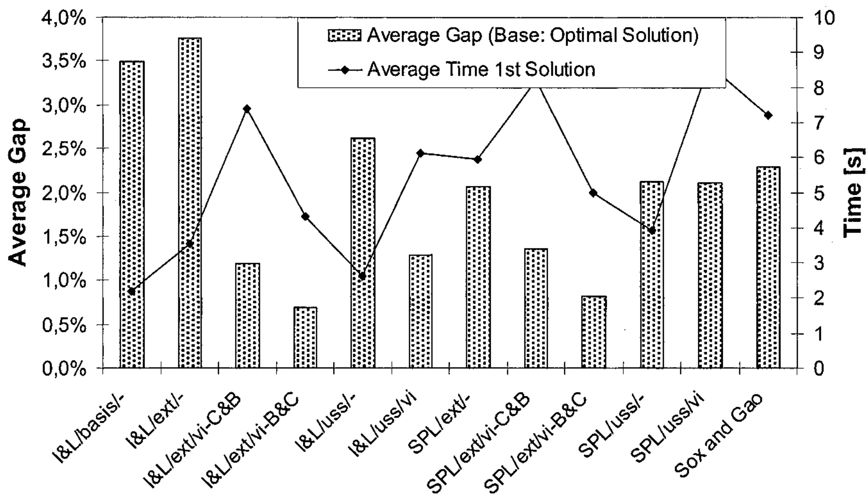


Fig. 7-1: Average gaps (lower bound: optimal solution) after 20s and average time to find a first feasible solution of different model formulations (test set *TT*1).

Fig. 7-2 shows how long it takes for different model formulations to prove optimality to the test instances of test set *TT*1. Apparently, SPL/ext/vi-B&C performs best and proves optimality to 81 out of 116 test instances within the time limit of 3 minutes. The benchmark model formulations by Sox and Gao (Gopalakrishnan) prove optimality to only 49 (2) test instances within the same time limit.

Test Sets *TT*2–*TT*9

Table 7-13 and Table 7-14 help to analyze the solution capabilities of the different model formulations for test sets *TT*2–*TT*9. In Table 7-13 the number of test instances for which a model formulation found the best solution is given for each test set. A time limit of 60s is imposed, but the first solution found thereafter is chosen, if none could be found within the time limit. The numbers in parenthesis indicate, how the result changes, if these solutions found after the time limit are excluded. Almost no solution is obtained within the time limit for test sets *TT*4 and *TT*6, which are the test sets with the largest number of periods (30) and products (12 and 24), but as argued above[340] the test sets with large numbers of products are not well suited for the CLSPL anyway.

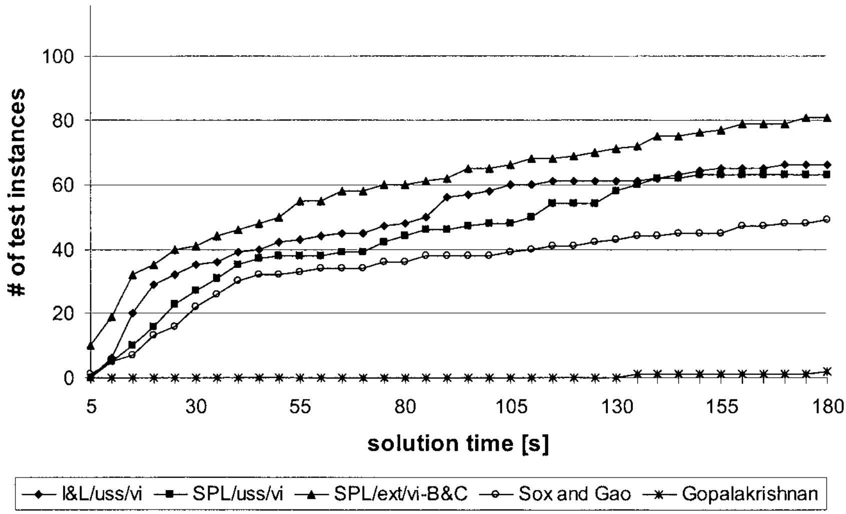


Fig. 7-2: Number of test instances solved to optimality over time (different model formulations, test set *TT*1).

Table 7-13: Number of test instances (total: 5 (*TT*2–*TT*6); 180 (*TT*7–*TT*9)) a model formulation found the best solution (# best) or no solution (# none) within the time limit of 60s.

	*TT*2		*TT*3		*TT*4		*TT*5		*TT*6	
	# best	# none	# best	# none	# best	# none	# best	# none	# best	# none
I&L/ext/vi-C&B	0	3	0	0	0	5	0	5	0	5
I&L/ext/vi-B&C	0	0	0	0	0	5	0	1	0	5
I&L/uss/vi	0	2	0	0	0	5	0	5	0	5
SPL/ext/–	0	0	0	0	0	5	0	0	0	5
SPL/ext/vi-C&B	2 (1)	3	3	0	2 (0)	5	1	0	2 (0)	5
SPL/ext/vi-B&C	3	1	5	0	3 (0)	4	2	0	4 (0)	5
SPL/uss/vi	0	4	0	0	0	5	2	1	1 (0)	5

	*TT*7		*TT*8		*TT*9	
	# best	# none	# best	# none	# best	# none
I&L/ext/vi-C&B	42 (41)	77	39 (38)	129	44 (42)	136
I&L/ext/vi-B&C	51 (49)	22	45 (44)	99	47 (45)	125
I&L/uss/vi	40	70	35 (34)	136	36 (34)	142
SPL/ext/–	54 (53)	6	54 (51)	87	48 (43)	112
SPL/ext/vi-C&B	90 (71)	57	90 (58)	100	105 (57)	117
SPL/ext/vi-B&C	107 (93)	43	102 (75)	79	105 (70)	103
SPL/uss/vi	61 (49)	55	77 (50)	111	76 (42)	131

Numbers in parenthesis indicate the result (if altered) if the first solution is taken for those instances exceeding the time limit.

340 See section 7.1.1 and 7.1.2.

Remarkably, all best solutions to test instances *TT*2–*TT*6 have been obtained by model formulations with an SPL basis and valid inequalities. Regarding test sets *TT*7–*TT*9, the advantage of these model formulations compared to the others seems to increase, if the number of products increases. These results are confirmed by Table 7-14 which provides the average gaps of the solutions found after 60s based on LB^{XLP}, the lower bound obtained by XpressMP after automatic cut generation. The average gap tends to be lower for test sets with a higher number of products, which is possibly explained by the reasoning that the relative portion of setup costs that is to be saved due to the linking feature of the CLSPL decreases, if more products are to be scheduled.

Table 7-14: Average gaps (lower bound: LB^{XLP}) of different model formulations for test sets *TT*2–*TT*9 (time limit: 60s or first solution).

Model formulation	*TT*2	*TT*3	*TT*4	*TT*5	*TT*6
I&L/ext/vi-C&B	13.65%	6.70%	11.41%	5.70%	9.20%
I&L/ext/vi-B&C	13.57%	6.82%	11.65%	5.13%	9.93%
I&L/uss/vi	19.52%	8.96%	12.49%	5.85%	9.56%
SPL/ext/–	14.02%	6.25%	7.61%	3.23%	4.29%
SPL/ext/vi-C&B	4.65%	0.72%	1.07%	0.79%	0.51%
SPL/ext/vi-B&C	3.71%	0.69%	1.22%	0.49%	0.35%
SPL/uss/vi	8.44%	2.12%	1.91%	1.08%	0.60%

	*TT*7	*TT*8	*TT*9
I&L/ext/vi-C&B	11.35%	8.35%	7.06%
I&L/ext/vi-B&C	11.17%	8.72%	7.59%
I&L/uss/vi	12.58%	8.86%	7.35%
SPL/ext/–	11.75%	5.97%	4.06%
SPL/ext/vi-C&B	3.15%	1.64%	1.19%
SPL/ext/vi-B&C	3.14%	1.68%	1.20%
SPL/uss/vi	4.16%	2.08%	1.33%

Test Sets *TL*1–*TL*3

In Table 7-15 and Table 7-16 the same information as in Table 7-13 and Table 7-14 is provided, but now for test sets *TL*1–*TL*3, which are better suited to assess the computational performance of the CLSPL. Again, model formulations with valid inequalities outperform those without. Furthermore, a feasible solution is obtained for all test instances with all model formulations for test set *TL*1 and for almost all for test set *TL*2. For these two test sets, in addition to the best model formulations for test sets *T*1–*T*9, also model formulation I&L/ext/vi-C&B provides reasonable results. Interestingly, test set *TL*2 seems to be more difficult (based on the observed average gaps) for the model formulations with an SPL basis, while the difficulty seems to increase with the number of products (4, 6 and 8 for test sets *TL*1, *TL*2 and *TL*3) for model formulations with an I&L basis.

Table 7-15: Number of test instances (total: 180) a model formulation found the best solution (# best) or no solution (# none) within the time limit of 60s.

	*TL*1		*TL*2		*TL*3	
Model formulation	# best	# none	# best	# none	# best	# none
I&L/ext/vi-C&B	94	0	75	4	57 (55)	13
I&L/ext/vi-B&C	76	0	75	0	66	5
I&L/uss/vi	73	0	64 (63)	2	52	14
SPL/ext/–	66	0	70	2	73	5
SPL/ext/vi-C&B	75	0	68 (65)	6	81 (66)	51
SPL/ext/vi-B&C	98	0	79	4	103 (97)	18
SPL/uss/vi	63	0	67	4	58 (56)	26

Numbers in parenthesis indicate the result (if altered) if the first solution is taken for those instances exceeding the time limit.

Table 7-16: Average gaps (lower bound: LB^{XLP}) of different model formulations for test sets *TL*1–*TL*3 (time limit: 60s or first solution).

Model formulation	*TL*1	*TL*2	*TL*3
I&L/ext/vi-C&B	10.78%	13.23%	21.49%
I&L/ext/vi-B&C	20.52%	23.07%	20.28%
I&L/uss/vi	16.80%	20.34%	26.46%
SPL/ext/–	28.59%	32.77%	26.26%
SPL/ext/vi-C&B	11.35%	15.23%	10.85%
SPL/ext/vi-B&C	11.65%	14.42%	10.33%
SPL/uss/vi	15.96%	19.11%	15.15%

Finally, in Fig. 7-3 to Fig. 7-5 average gaps of the different model formulations are shown. The upper bound is the solution obtained within the time limit of 60s or the first solution, if it has been found later. The lower bounds used are LB^{XLP}, the lower bound obtained after the automatic cut generation of XpressMP, the bound obtained by the branch-and-bound search after 60s or at the time the first solution has been found (whichever is later) (best bound), as well as the best solution obtained by any of the model formulations tested (best solution). The difference between the first two bounds shows how much of the gap is closed within the first minute of computational time. This portion is clearly bigger for test set *TL*1 (Fig. 7-3) than for the other two test sets *TL*2 (Fig. 7-4) and *TL*3 (Fig. 7-5).

Furthermore, it is interesting to note that the ranking of the different model formulations changes for the different test sets. In Fig. 7-3 (test set *TL*1) model formulation I&L/ext/vi-C&B seems to outperform the other model formulations. Based on the Wilcoxon matched-pairs signed-ranks tests the solutions obtained by this model formulation are significantly better than even those of model formulations SPL/ext/vi-C&B and SPL/ext/vi-B&C, which come in second and third, at confidence levels of 99%. However, this result is reversed for test sets *TL*2 and *TL*3. For both test sets, SPL/ext/vi/B&C outperforms I&L/ext/vi/C&B significantly based on the same statistical test as above. The reason for this difference is presumably the different size of the test instances in each test set. The I&L model formulation seems to work much better with smaller test instances.

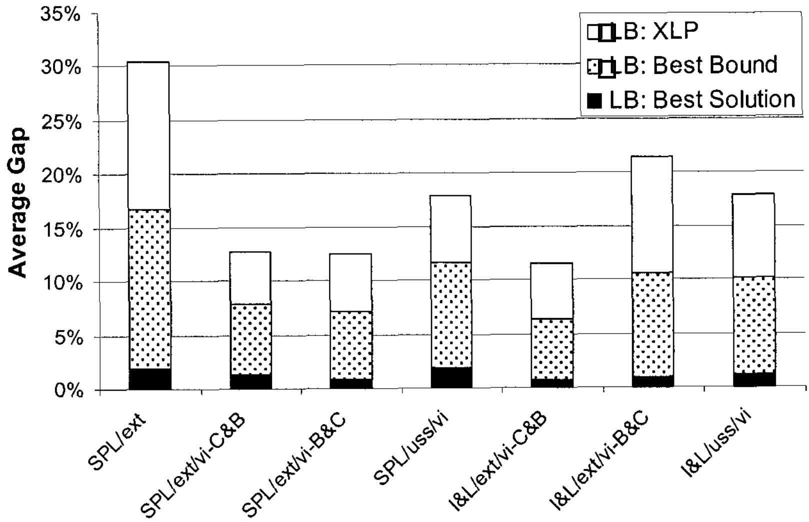


Fig. 7-3: Average gaps (to different lower bounds) of different model formulations (test set *TL*1, time limit: 60s or first solution).

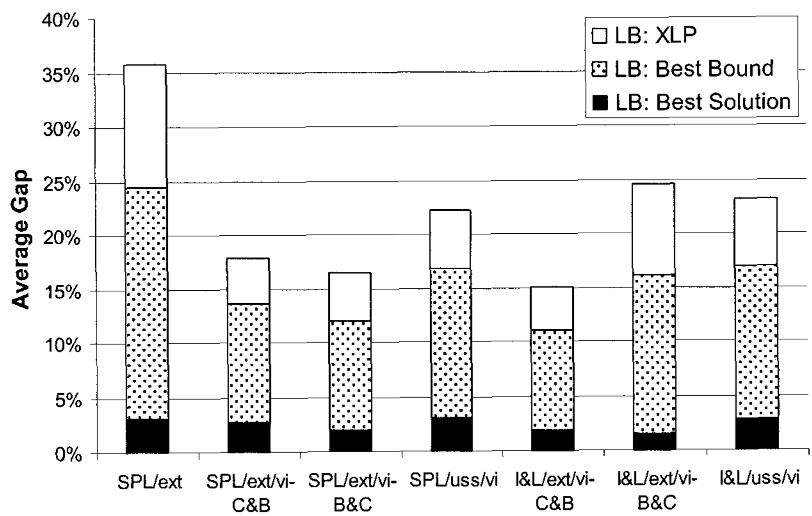


Fig. 7-4: Average gaps (to different lower bounds) of different model formulations (test set *TL*2, time limit: 60s or first solution).

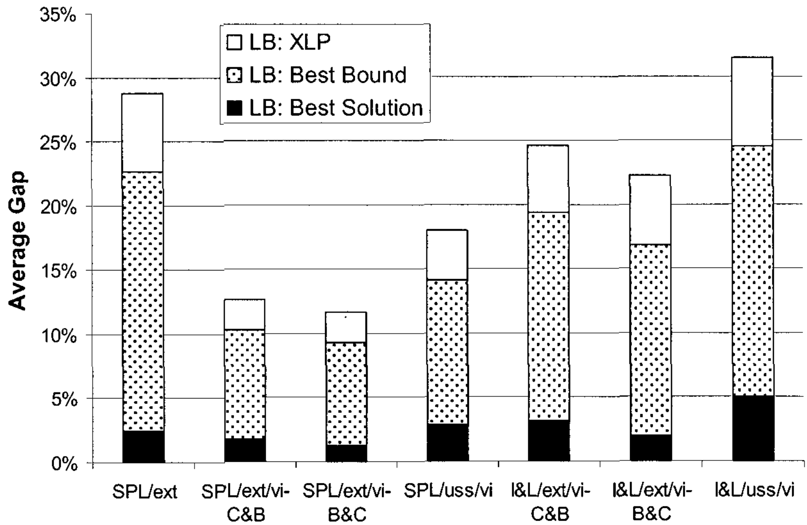


Fig. 7-5: Average gaps (to different lower bounds) of different model formulations (test set *TL3*, time limit: 60s or first solution).

Decomposition Heuristic

Thus far, the different model formulations have been compared. Lastly, the performance of the temporal decomposition heuristic presented in section 6.2 will be assessed. As the model formulations itself provide a reasonable solution quality, if the underlying decision problem is not too big in terms of number of products and periods, the heuristic is intended to increase solution speed considerably without loosing too much solution quality, if the solution quality of the model deteriorates due to its matrix size.

Table 7-17 provides a comparison of solution quality (based on the average gap between the best solution found and the lower bound after automatic cut generation of model SPL/ext/vi-C&B) and solution speed. The best model formulation, the decomposition heuristic and the tabu-search heuristic by Gopalakrishnan et al. (2001) are to be compared.[341] The decomposition heuristic has been run with parameter combination 6/2/2, a time limit of 15 seconds and no anticipation of setup times after the rolling window (option "*min*"). Computational times are only an indicator here, because different hardware has been used by Gopalakrishnan et al. (2001) compared to our tests. Gaps are much lower for either the model formulation or the decomposition heuristic compared to the tabu-search heuristic by Gopalakrishnan et al. (2001). As their solutions and lower bounds have not been available, a further analysis, whether the difference stems from better solutions or

[341] Data for the tabu-search heuristic by Gopalakrishnan et al. (2001) are taken from their paper. Cf. Gopalakrishnan et al. (2001) pp. 861-862.

better bounds is not possible. However, although the decomposition heuristic provides slightly worse solutions compared to the model formulation, it takes a much shorter amount of time to arrive at these solutions.

Table 7-17: Comparison of model formulation SPL/ext/vi-B&C, decomposition heuristic and tabu-search heuristic of Gopalakrishnan et al. (2001). Gaps (except Gopalakrishnan et al. (2001)) are based on $LB^{XLP(SPL/ext/vi)}$.

Test sets *TT*	SPL/ext/vi-B&C (60s or first solution)			Decomposition heuristic (6/2/2, 15s, *min*)		Gopalakrishnan et al. (2001) [Pentium 3, 550 MHz]	
	Gap		Time [s]	Gap	Time [s]	Gap	Time [s]
1, 2	1.9%	(1)	41.5	3.0%	6.3	27.8%	
3, 4	1.0%	(4)	65.6	1.9%	17.9	13.9%	} 20.8
5, 6	0.4%	(5)	122.4	1.5%	32.6	6.0%	
7–9	2.0%	(225)	155.4	4.1%	23.8	12.4%	81.7

The performance of the temporal decomposition heuristic will be analyzed in some more detail based on test sets *TL*1–*TL*3, which are better suited for the CLSPL as has been argued above. A crucial factor must be the choice of a good parameter combination, i.e. the choice of the size of the rolling windows. Different parameter combinations are compared in Table 7-18. Here, and in all subsequent tables, corresponding data for model formulation SPL/ext/vi-B&C are given as reference values. Apart from test set *TL*1 the decomposition heuristic finds on average better solutions than the reference model formulation for at least one parameter combination (6/2/2). To do so, the decomposition heuristic only needs a fraction of the time compared to the reference model formulation.

The parameter combination without overlap of rolling windows (4/0/0) shows the worst average solution quality (Gap) and worst case performance (Max). The solution quality is significantly[342] worse than for all other parameter combinations. Still, it outperforms almost all other model formulations (Table 7-16). Overlapping rolling windows as well as relaxing integrality constraints within this overlap (4/0/2 and 4/2/2) improves average solution quality as well as worst case behavior. However, the computational effort almost doubles, if integrality constraints are not relaxed. Increasing the size of the rolling windows while retaining an overlap with relaxed integrality constraints (6/2/2) seems to be the best choice balancing the gain in solution quality with the loss in solution speed. This parameter combination shows a significantly superior solution quality compared to all other combinations depicted in Table 7-18.

In Table 7-19 different variants to anticipate setup times in periods after the rolling window are compared. Although the best solution quality (Gap) is reported, if the anticipation deducts a product's setup time from available capacity in each period with positive demand (anticipation of setup times: *max*), this stems from the fact that approximately 20% of test instances of each test set could not be

342 Again the Wilcoxon matched-pairs signed-ranks test is used here and in the following paragraphs at confidence levels of 99% to obtain conclusions regarding the different parameter combinations of the temporal decomposition heuristic.

solved with this kind of anticipation. Often even the first rolling window is infeasible rendering the whole test instance infeasible for the heuristic.

If setup times are anticipated by option "*mean*", average and worst case performance are slightly improved versus variant "*min*", if parameter combination 4/2/2 is used. However, the opposite is true for parameter combination 6/2/2, but neither of these two differences are significant.

Table 7-18: Performance of decomposition heuristic for test sets *TL*1–*TL*3 and different sizes of the rolling window (time parameter: 15s; anticipation of setup times: *min*). Gap and maximum deviation are based on $LB^{XLP(SPL/ext/vi)}$ used as a lower bound.

Parameter combination	*TL*1 Gap	*TL*1 Max	*TL*1 Time[s]	*TL*2 Gap	*TL*2 Max	*TL*2 Time [s]	*TL*3 Gap	*TL*3 Max	*TL*3 Time[s]
SPL/ext/vi-B&C	10.7%	32.4%	50.9	14.4%	51.7%	128.6	10.3%	48.4%	286.7
4 / 2 / 2	13.3%	45.7%	2.5	14.6%	46.5%	3.6	10.1%	29.5%	5.3
4 / 0 / 0 [(a)]	14.9%	52.7%	1.8	17.1%	57.6%	2.6	13.4%	44.1%	3.8
4 / 0 / 2	12.6%	42.2%	3.5	14.6%	44.2%	5.1	10.2%	26.9%	7.5
6 / 2 / 2	11.2%	39.6%	3.6	12.9%	37.6%	6.2	9.5%	32.4%	9.5

[(a)] For two instances of test sets *TL*2 and *TL*3 no solution has been obtained with this parameter combination (4/0/0).

Table 7-19: Performance of decomposition heuristic for test sets *TL*1–*TL*3 and different anticipations of setup times (time parameter: 15s). Gap and maximum deviation are based on $LB^{XLP(SPL/ext/vi)}$ used as a lower bound.

Anticipation	Parameter combination	*TL*1 Gap	*TL*1 Max	*TL*1 Time [s]	*TL*2 Gap	*TL*2 Max	*TL*2 Time [s]	*TL*3 Gap	*TL*3 Max	*TL*3 Time [s]
SPL/ext/vi-B&C		10.7%	32.4%	50.9	14.4%	51.7%	128.6	10.3%	48.4%	286.7
min	4 / 2 / 2	13.3%	45.7%	2.5	14.6%	46.5%	3.6	10.1%	29.5%	5.3
mean	4 / 2 / 2	13.0%	35.2%	2.6	14.5%	43.8%	3.7	9.9%	30.1%	5.4
max[(a)]	4 / 2 / 2	(10.6%)	(28.7%)	2.0	(12.1%)	(32.7%)	2.7	(8.8%)	(29.0%)	3.9
min	6 / 2 / 2	11.2%	39.6%	3.6	12.9%	37.6%	6.2	9.5%	32.4%	9.5
mean	6 / 2 / 2	11.8%	40.7%	3.8	13.0%	31.0%	6.5	9.7%	28.2%	9.7
max[(b)]	6 / 2 / 2	(9.7%)	(25.5%)	3.0	(11.3%)	(31.0%)	4.9	(8.5%)	(32.5%)	7.3

[(a)] For many test instances no feasible could be obtained with this anticipation of setup times (*TL*1: 35, *TL*2: 41, *TL*3: 39)

[(b)] For many test instances no feasible could be obtained with this anticipation of setup times (*TL*1: 29, *TL*2: 34, *TL*3: 34).

As solutions are obtained very quickly with the temporal decomposition heuristic, in Table 7-20 the effect of the time parameter is analyzed. Increasing it from 5s to 15s and from 15s to 30s adds only marginal additional time that is actually used (test set *TL*1). However, the first increase improves solution quality significantly. For test set *TL*2, the first increase (on average) leads to almost one second more computational time used, again accompanied with a significant increase in solution quality. Only for test *TL*3 both increases in computational time lead to significant increases in solution quality. However, the imposed time limit is never exceeded and computational times are only 11.2s on the average.

Table 7-20: Performance of decomposition heuristic for test sets *TL*1–*TL*3 and different time parameters (parameter combination: 6/2/2). Gap and maximum deviation are based on $LB^{XLP(SPL/ext/vi)}$ used as a lower bound.

Time parameter	Anticipation	*TL*1 Gap	*TL*1 Max	*TL*1 Time [s]	*TL*2 Gap	*TL*2 Max	*TL*2 Time [s]	*TL*3 Gap	*TL*3 Max	*TL*3 Time [s]
SPL/ext/vi-B&C		10.7%	32.4%	50.9	14.4%	51.7%	128.6	10.3%	48.4%	286.7
5	*min*	11.8%	39.6%	3.5	14.8%	43.7%	5.4	11.1%	37.2%	7.2
5	*mean*	12.6%	40.7%	3.7	15.0%	39.7%	5.5	11.3%	36.4%	7.4
15	*min*	11.2%	39.6%	3.6	12.9%	37.6%	6.2	9.5%	32.4%	9.5
15	*mean*	11.8%	40.7%	3.8	13.0%	31.0%	6.5	9.7%	28.2%	9.7
30	*min*	11.2%	39.6%	3.6	12.9%	37.6%	6.3	8.7%	25.7%	10.9
30	*mean*	11.8%	40.7%	3.8	12.9%	31.0%	6.5	8.8%	23.3%	11.2

Finally, Table 7-21 analyzes the effect of bonuses. As indicated in section 6.2.3 bonuses are intended to reduce the so-called planning horizon effect. Bonuses are used here without the component valuing the loss of capacity due to setup times after the rolling window. For parameter combination 4/2/2 bonuses decrease the average gap for all test sets significantly. Especially, the worst case performance (column "Max") is improved. However, the effect of bonuses diminishes, if parameter combination 6/2/2 is used. There, the average gap even deteriorates for test set *TL*3. However, while parameter combination 6/2/2 provides the best solution quality with bonuses and no anticipation of setup times for test sets *TL*1 and *TL*2, for test set *TL*3 parameter combination 4/2/2 provides the best solution quality with the same anticipation used.

Table 7-21: Performance of decomposition heuristic for test sets *TL*1–*TL*3 and different anticipations of setup times including bonuses (time parameter: 15s). Gap and maximum deviation are based on $LB^{XLP(SPL/ext/vi)}$.

Anticipation	Parameter combination	*TL*1 Gap [%]	*TL*1 Max [%]	*TL*1 Time [s]	*TL*2 Gap [%]	*TL*2 Max [%]	*TL*2 Time [s]	*TL*3 Gap [%]	*TL*3 Max [%]	*TL*3 Time [s]
SPL/ext/vi-B&C		10.7	32.4	50.9	14.4	51.7	128.6	10.3	48.4	286.7
min	4 / 2 / 2	13.3	45.7	2.5	14.6	46.5	3.6	10.1	29.5	5.3
min+bonus	4 / 2 / 2	12.2	32.6	3.0	13.7	37.3	4.2	9.5	21.4	6.2
mean	4 / 2 / 2	13.0	35.2	2.6	14.5	43.8	3.7	9.9	30.1	5.4
mean+bonus	4 / 2 / 2	12.5	28.3	2.9	13.8	35.9	4.4	9.5	21.7	6.4
min	6 / 2 / 2	11.2	39.6	3.6	12.9	37.6	6.2	9.5	32.4	9.5
min+bonus	6 / 2 / 2	11.1	27.6	4.3	12.8	33.8	7.5	10.0	31.0	10.4
mean	6 / 2 / 2	11.8	40.7	3.8	13.0	31.0	6.5	9.7	28.2	9.7
mean+bonus	6 / 2 / 2	11.7	40.7	4.4	13.1	29.5	7.7	10.3	35.8	10.7

Summarizing, regarding the computational performance all improvements compared to the basic version of CLSPL have had positive effects. Both, reformulations (SPL reformulation and extended model formulation) and the addition of valid inequalities helped to increase the computational performance of the proposed CLSPL model formulation. Compared to the benchmarks from literature, both the proposed model formulations as well as the temporal decomposition heuristic proved to be very effective. Furthermore, the decomposition heuristic improved the quality of solutions compared to the best model formulation proposed for test sets *TL*1–*TL*3 using far less computational time. Regarding their customization it proved worthwhile to include an overlap of rolling windows and bonuses to overcome the so-called planning horizon effect. Setup times after the rolling windows have been anticipated best by options "*min*" and "*mean*".

7.2 Time Continuity – Lot Sizes

7.2.1 Test Sets and Benchmark

In this section almost the same test sets which have been introduced in the preceding section 7.1.1 as test sets *TL*1 and *TL*2 will be used. These test sets will be enriched by minimal and maximal lot sizes as well as different batch sizes to pose corresponding restrictions. Furthermore, their demand profile will be altered slightly to make them feasible at least for the basic PLSP. The most test instances of test sets *TL*1 and *TL*2 are not feasible for the PLSP, because demand for all products generally starts in the first period and the PLSP can at most set up one product per period.

The demand profiles are changed according to the following procedure to yield test set *TC*1 based on test set *TL*1. Demands of products j=2..J in periods t=1..j–1 are shifted to the last period within the planning interval (T). Symmetrically, this is done to yield test set *TC*2 based on test set *TL*2. Demands of products j=J–1..1 in periods t=1..J–j are shifted to the last period within the planning interval (T).

Furthermore, to generate also a test set with a larger number of periods to assess the computational performance of the temporal decomposition heuristic, test set *TC*4 (*TC*5) is generated by doubling the number of periods in test set *TL*1 (*TL*2) and afterwards applying the procedure to make the test instances feasible at least for the PLSP which has been used to generate test set *TC*1 (*TC*2) based on *TL*1 (*TL*2).

Lot size restrictions are imposed on all products. The batch size (bs_j) has been chosen to be 250 units for all products, if not stated otherwise. The minimum lot size ($minlot_j$) is $4 \cdot bs_j$ ($1 \cdot bs_j$) for products j=1, 2 (j=any other product). This also reflects the different characteristic of the demand profiles of these two product groups. The maximum lot size ($maxlot_j$) is 3,000 for all products. This might have been the result of the following planning scenario (test set *TC*1): Four products with different demand profiles (two products face a fourfold higher demand than the other two products) require treatment on one bottleneck resource. This resource operates only in batch mode with a batch size of 250 units. The resource needs to be cleaned (requires a setup operation) after having processed at most 12 batches.

In two other scenarios the batch size is reduced to bs_j=125 (bs_j=50) to calculate the cost advantage, if the equipment allows to produce in smaller batches. Furthermore, a scenario with a smaller maximum lot size ($maxlot_j$=1,500) is defined to analyze model formulations and solutions which allow to produce consecutive lots of the same product.

As the model formulation by Kallrath (1999)[343] is the only model formulation with similar properties compared to the PLSP model formulation with the lot size extension, it is chosen as a benchmark here. The benchmark formulation requires to determine a parameter α_j beforehand, which is the number of lots that are maximally produced of product j within the planning interval. Consequently, if α_j is chosen too low, the optimal solution cannot be found, while on the other hand the matrix size is inflated unnecessarily, if α_j is chosen too high. Here, for test set *TC*1 α_j has been set to six for all products in all test instances after extensive preliminary tests. Furthermore, although Kallrath (1999) has provided a different basic PLSP model formulation in his paper,[344] the one used throughout this thesis supplemented by valid inequalities is taken,[345] which gives much better results.

7.2.2 Analysis of Solutions

To analyze the behavior of model formulations with the lot size extension, the following notation is introduced (Table 7-22). The first field represents the basic model formulation, which is either the PLSP or the CLSPL. The second field represents the different lot size restrictions present (none, maximal lot sizes, minimal lot sizes, production in integer multiples of a batch size, batch flow scenario, consecutive lots of the same product allowed). Finally, the third field describes the use of valid inequalities or indicates the use of the preprocessing procedure outlined for the batch flow scenario in section 6.1.2.1. Valid inequalities are either applied at the root node only (cut-and-branch approach, "vi-C&B") or throughout the branch-and-bound search. The latter is indicated by "vi-B&C". Again, the model cut feature of XpressMP is used here. E.g., the abbreviation "PLSP/minmax/vi-B&C" describes a model formulation using the PLSP as the basic model formulation, obeying restrictions on minimal and maximal lot sizes and applying valid inequalities via the model cut feature of XpressMP.

Difficulty of Different Restrictions

From Fig. 7-6 differences regarding computational tractability of model formulations with different restrictions imposed on lot sizes become obvious. It is depicted, how many test instances (test set *TC*1; total: 180) have been solved to proven optimality over time. Apparently, the most easy model formulation is the basic model PLSP with no additional constraints. All test instances are solved to optimality within the time limit of three minutes. The second easiest case are minimal and maximal lot size restrictions. The constraints associated with these restrictions do not contain any additional binary or integer variables. Third comes the model formulation in which batch size restrictions are imposed on production lots, and fourth the model formulation which contains all of the above restrictions

---

343 Cf. Kallrath (1999) pp. 331-334. The model formulation can be found in the appendix.

344 Cf. Kallrath (1999) p. 332.

345 See sections 2.3.3 and 6.1.1.1. Valid inequalities (6-3) and (6-4) are defined as regular constraints here, because this setting provides the best outcome for the benchmark.

at the same time. Most difficult seems to be the batch flow scenario. However, this ranking changes somewhat, when the preprocessing procedure outlined in section 6.1.2.1 is applied for the batch flow scenario (Fig. 7-11 and Table 7-32).

Table 7-22: Notation of different model formulations.

Parameter	Value	Comment
1	PLSP	Basic model formulation *PLSP* (I&L)
	CLSPL	Basic model formulation *CLSPL* (I&L)
2	–	No further restrictions
	max	With maximal lot size restrictions (see section 6.1.2.1)
	minmax	With minimal and maximal lot size restrictions (see section 6.1.2.1)
	bat	With batch size restrictions (see section 6.1.2.1)
	all	With minimal, maximal and batch size restrictions (see section 6.1.2.1)
	+flow	Additionally with the batch flow scenario (see section 6.1.2.1)
	+cc1	Additionally with consecutive campaigns of the same product allowed (see section 6.1.2.1, first modeling approach)
	+cc2	Additionally with consecutive campaigns of the same product allowed (see section 6.1.2.1, second modeling approach)
	+K	Additional constraint to allow for a fair comparison with the Kallrath (1999) model formulation (see section 7.2.3)
3	vi-C&B	Valid inequalities (see section 6.1.2.1), *C*ut*&B*ranch approach (=valid inequalities only applied in the root node, see section 5.2.1.2)
	vi-B&C	Valid inequalities (see section 6.1.2.1), *B*ranch*&C*ut approach (=model cut feature of XpressMP, see section 5.2.1.4)
	+pp	Preprocessing procedure applied (see section 6.1.2.1)
	+YB	Variables Y_{jt} defined as binary variables (see section 6.1.2.1)

In Table 7-23 the aforementioned results are confirmed and provided in some more detail. If the lot size extension can be modeled without extra binary or integer variables (max and minmax), all test instances are solved to proven optimality within the time limit, first solutions are found fairly quick (in less than three seconds on the average) and also the optimal solutions are obtained in a reasonable amount of time. If batch size restrictions are present, it takes a bit longer to find a first feasible solution and only two thirds of all instances have been proven optimal within the time limit. Finally, the batch flow scenario takes the longest to find a first feasible solution and proves optimality to a bit less than one half of the test instances of test set *TC*1.

Table 7-24 analyzes the average and maximum cost increase compared to the optimal PLSP solution based on test instance characteristics. As described above[346] the 180 test instances of test set *TC*1 originate from test set *TT*7 defined by Trigeiro et al. (1989).[347] These have been generated using three capacity utilization profiles (low, med. and high), three TBO profiles (low, med. and high), two different coefficients of variation for demands as well as two setup time profiles. For each of these 3 x 3 x 2 x 2 = 36 combinations five different demand

[346] See sections 7.1.1 and 7.2.1.

[347] Cf. Trigeiro et al. (1989) pp. 358-362.

series have been randomly generated to yield the 180 test instances. Consequently, one can evaluate the effect of these characteristics by taking the aggregation scheme into consideration.

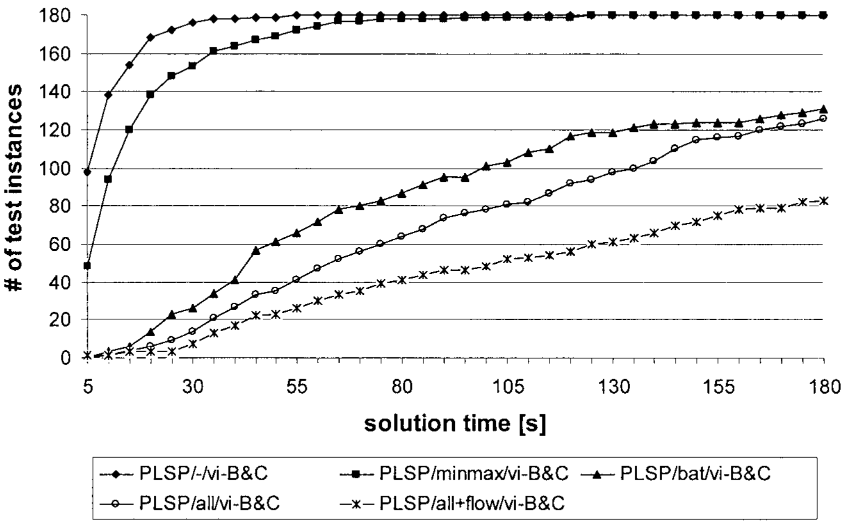


Fig. 7-6: Number of test instances solved to optimality over time (different lot size restrictions, test set *TC*1).

Table 7-23: Solution speed of model formulations with different lot size restrictions for test sets *TC*1.

Number of test instances for which	PLSP/ * /vi-B&C					
* =	–	max	minmax	bat	all	all+flow
- a feasible solution is found within 15s	180	180	180	178	177	164
- a feasible solution is found within 30s	180	180	180	179	180	175
- a feasible solution is found within 60s	180	180	180	180	180	177
- a feasible solution is found within 90s	180	180	180	180	180	177
- a feasible solution is found within 180s	180	180	180	180	180	179
- optimality is proven within 180s	180	180	180	131	126	83
Average time (in seconds) to						
- find a first solution	1.17	2.14	2.72	5.37	6.12	10.01
- find an optimal solution[(a)]	4.86	8.28	11.29	50.44	61.70	57.06
- prove optimality[(a)]	6.45	10.36	14.97	65.83	84.56	87.69

[(a)] Only for test instances with proven optimality

From Table 7-24 it becomes clear that the TBO profile has the biggest impact on cost increases compared to optimal PLSP solutions. TBO profile "high" means that setup costs are expensive compared to inventory holding costs or, equivalently, holding inventory is relatively cheap. Restrictions on lot sizes (e.g., minimal lot sizes or batch size restrictions) yield ceteris paribus (here: the setup pattern stays the same) higher stocks. Thereby, the cost increase has to be bigger for TBO profile "low" than for TBO profile "high", which definitely is the case here. Of course, the ceteris paribus assumption is not valid here, because the setup pattern will change to offset (or at least alleviate) the cost increase, but this is one explanation for the impact of the TBO profile on costs here.

Table 7-25 provides average and maximum gaps for the same model formulations and at the same level of detail as Table 7-24. This data is provided here, because the solutions which have been used to derive Table 7-24 have not been optimal for all model formulations. So, the interpretation obtained from Table 7-24 might have been disguised due to bad solutions. However, this is not the case, as the gaps are fairly low (Table 7-25).

Table 7-24: Average and maximum percentage deviation from optimal PLSP solutions of different model formulations (time limit: 3 minutes) for test set *TC*1.

Average	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/max/vi-B&C	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
PLSP/minmax/vi-B&C	4.3	5.0	4.1	8.3	4.4	0.7	3.9	5.1	4.4	4.5
PLSP/bat/vi-B&C	25.9	26.7	26.8	41.9	25.1	12.5	25.8	27.1	26.8	26.1
PLSP/all/vi-B&C	29.2	29.7	29.5	47.3	28.1	12.9	28.6	30.2	29.6	29.3
PLSP/all+flow/vi-B&C	41.5	40.6	39.6	64.6	38.5	18.5	40.0	41.1	40.3	40.8
Maximum	**Capacity utilization**			**TBO profile**			**Demand variation**		**Setup time length**	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/max/vi-B&C	0.6	0.6	0.8	0.0	0.0	0.8	0.0	0.8	0.2	0.8
PLSP/minmax/vi-B&C	27.1	19.8	15.1	27.1	11.8	4.2	19.8	27.1	27.1	17.2
PLSP/bat/vi-B&C	56.1	61.9	51.1	61.9	32.6	19.5	51.1	61.9	61.9	51.4
PLSP/all/vi-B&C	63.5	65.7	57.7	65.7	36.6	19.5	63.1	65.7	65.7	63.1
PLSP/all+flow/vi-B&C	85.1	88.3	70.1	88.3	49.0	30.5	86.6	88.3	88.3	85.1

Table 7-25: Average and maximum gap [%] (upper bound: best solution found within a time limit of 3 minutes; lower bound: bound after 3 minutes) for test set *TC*1.

Average	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/–/vi-B&C	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PLSP/max/vi-B&C	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PLSP/minmax/vi-B&C	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PLSP/bat/vi-B&C	0.9	1.1	2.5	0.0	0.4	4.2	1.8	1.2	1.5	1.5
PLSP/all/vi-B&C	0.9	1.6	2.7	0.0	0.7	4.5	2.3	1.2	1.6	1.9
PLSP/all+flow/vi-B&C	3.4	3.6	9.2	0.8	4.5	10.8	5.7	5.0	6.4	4.4
Maximum	**Capacity utilization**			**TBO profile**			**Demand variation**		**Setup time length**	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/–/vi-B&C	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PLSP/max/vi-B&C	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PLSP/minmax/vi-B&C	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PLSP/bat/vi-B&C	7.7	7.7	17.2	0.0	6.3	17.2	9.8	17.2	17.2	12.2
PLSP/all/vi-B&C	6.9	8.9	13.5	2.9	7.3	13.5	13.5	10.2	13.5	11.5
PLSP/all+flow/vi-B&C	12.8	12.9	29.4	13.4	29.4	27.9	29.4	27.9	29.4	24.4

Although the cost increase compared to optimal PLSP solutions seems to be affected only by the TBO profile, the gaps seem to be not only dependent on the TBO profile, but also on the capacity utilization level (Table 7-25). The gaps, which are the gaps that remain for each test instance after three minutes of computational time (deviation of best solution and best bound after three minutes), thereby – in a sense – measure the difficulty of test instances. High capacity utilization levels as well as sharp tradeoffs between setup costs and inventory holding costs seem to make the problem more difficult. Moreover, test instances with low demand variation (stable demand) show gaps that are a bit higher than those with high demand variation.

Different Batch Sizes

Obviously, if additional restrictions on the lot size are present, it will generally be no longer possible to find the same solution as before (for the basic PLSP). Consequently, the optimal solution to the basic PLSP is a lower bound for all model formulations with additional restrictions (on lot sizes). Furthermore, this deviation from the optimal PLSP solution has an economic interpretation: It is the cost of obeying these lot size restrictions, e.g., the cost of producing in batches. Thereby, it is possible to calculate the cost of different scenarios (e.g., different resources which allow for different batch sizes) and use the outcome not only for scheduling, but also as a help for investment decisions. The cost decrease in inventory holding costs due to smaller batch sizes (or minimal lot sizes) may pay off an investment in a resource which allows for smaller batch sizes (minimal lot sizes).

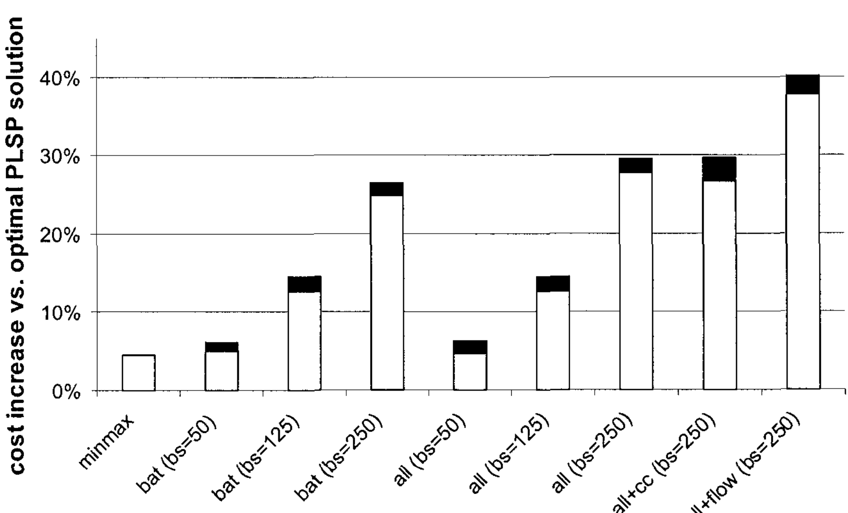


Fig. 7-7: Average cost increase vs. optimal PLSP solutions for different lot size restrictions (test set *TC*1, black: average gap after 3 minutes of computation).

In Fig. 7-7 the average cost increase compared to the optimal PLSP solution is shown for different lot size restrictions. As these model formulations have not been solved to optimality, it might be, that the cost increase stems from bad solutions. This is not the case here and the effect is depicted by the shaded (black) area of each bar, which represents the average gap between the solution used in this calculation and the lower bound obtained on this solution. This means the average cost increase of different lot size restrictions compared to the optimal PLSP solutions lies somewhere in the black area of each bar.

Minimal and maximal lot size restrictions have only a minor effect here (left bar). Different batch sizes on the other hand seem to have a big impact on costs. Furthermore, if consecutive lots of the same product are allowed, this results in a small cost reduction here (second and third bar from right).

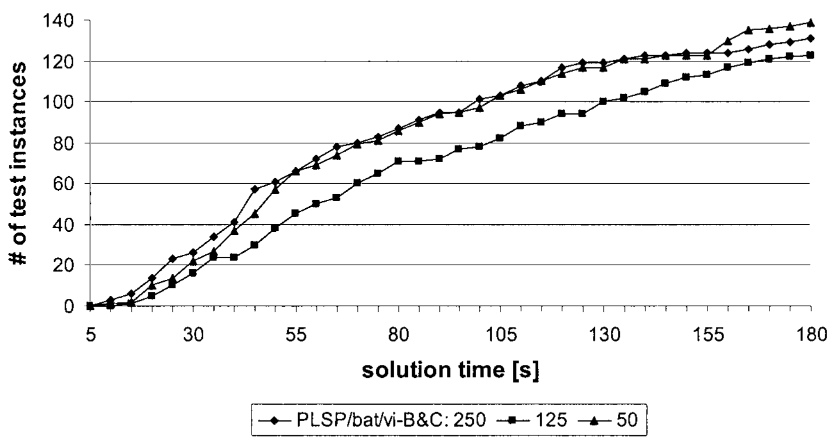


Fig. 7-8: Number of test instances solved to optimality over time (different batch sizes, test set *TC*1).

Table 7-26: Gap [%] (upper bound: best solution found within a time limit of 3 minutes; lower bound: LB^{XLP}) and cost comparison [%] to optimal PLSP solutions for different batch sizes (test set *TC*1).

	PLSP/–/	PLSP/bat/vi-B&C			PLSP/all/vi-B&C		
	vi-B&C	250	125	50	250	125	50
Gap (average)	4.8	22.4	23.5	22.4	23.7	23.5	22.4
Gap (standard deviation)	4.1	6.4	6.5	6.7	6.2	6.4	6.7
Gap (minimum)	0.0	8.9	12.1	12.2	8.0	12.6	12.3
Gap (maximum)	14.5	45.0	52.0	52.5	45.4	50.6	50.5
Cost increase vs. PLSP (average)	-	26.5	14.5	6.2	29.4	14.6	6.2
Cost increase vs. PLSP (std. dev.)	-	12.9	7.1	3.1	15.0	7.1	3.1
Cost increase vs. PLSP (minimum)	-	7.2	3.9	1.8	7.2	4.2	1.8
Cost increase vs. PLSP (maximum)	-	61.9	33.7	13.9	65.8	33.7	13.9

In Fig. 7-8 the effect of different batch sizes on the ability to obtain optimal solutions is analyzed. Apparently, the batch size seems to have only minor impact on the ability of the model formulation to prove optimality of the test instances. On the other hand, Table 7-26 shows that costs are strongly affected by the batch size. But there is an explanation to this phenomenon, because the bigger the batch size, the more stock will ceteris paribus lie in inventory.

Comparison of Basic Models PLSP and CLSPL

Table 7-27 shows the average percentage deviation in total costs between model formulations using the CLSPL as a basic model and those using the PLSP as the basic model. Theoretically, all values must be negative, because model formulations with the CLSPL as a basic model allow all solutions that are feasible with the PLSP as the basic model plus those, in which more than one product is set up in a single period. Therefore, solutions using the CLSPL must be cheaper than those using the PLSP. As not all solutions obtained have been proven optimal, sometimes (especially with TBO profile "high") model formulations using the PLSP produced better solutions within the time limit of 3 minutes.

Apparently, the cost difference between CLSPL and PLSP solutions is remarkably high for TBO profile "low". This is the expected result, because here setup costs are relatively low compared to inventory holding costs. Therefore, it makes sense to perform setup operations more frequently, what, of course, is easier integrated in plans based on the CLSPL as this basic model is not limited to one setup operation per period. Moreover, CLSPL plans are relatively cheaper for low capacity utilization. Also in this case, it seems advantageous to schedule more products in each period.

Furthermore, from Table 7-27 one can observe, that the cost advantage decreases the more lot size restrictions are present (all vs. max) and the stronger these restrictions are ($maxlot_j$=3,000 vs. $maxlot_j$=1,500).

Table 7-27: Average percentage deviation of objective function value (total cost) between CLSPL and corresponding PLSP solutions (time limit: 3 minutes) for test set *TC*1 with different maximum lot sizes.

maxlot = 3,000	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
CLSPL/max+cc2/vi-B&C	-24.3	-23.4	-21.0	-55.9	-12.1	-0.7	-23.4	-22.4	-22.9	-22.9
CLSPL/all+cc2/vi-B&C	-12.7	-9.5	-4.2	-21.3	-7.2	2.1	-9.1	-8.5	-7.6	-10.0

maxlot = 1,500	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
CLSPL/max+cc2/vi-B&C	-22.8	-22.2	-20.1	-56.3	-11.8	3.0	-22.3	-21.1	-21.7	-21.7
CLSPL/all+cc2/vi-B&C	-11.6	-8.7	-3.4	-22.5	-6.6	5.4	-8.3	-7.5	-6.1	-9.7

Finally, Table 7-28 provides the average number of setup operations per product and test instance. The numbers are given separately for those products with high demand and those products with low demand. As expected, especially products with low demand are set up much more often, if the CLSPL is used as a basic model. Moreover, the average number of setup operation increases, if the maximum lot size allowed decreases ($maxlot_j$=1,500 vs. $maxlot_j$=3,000).

Table 7-28: Number of setup operations in model formulations with different lot size restrictions (test set *TC*1).

Demand profile		high			low	
Setup operations	mean	min	max	mean	min	max
PLSP/–/vi-B&C	5.30	3	7	3.69	2	5
PLSP/all+cc2/vi-B&C	5.20	3	7	3.60	2	5
PLSP/all+cc2/vi-B&C [(a)]	5.74	4	7	3.71	2	5
CLSPL/minmax/vi-B&C	5.31	3	8	5.37	2	8
CLSPL/bat/vi-B&C	7.05	3	13	6.04	3	10
CLSPL/all/vi-B&C	5.57	3	8	5.80	3	10

[(a)] $maxlot_j$=1,500

Summarizing, the analysis of solutions revealed the different computational effort associated with different restrictions on lot sizes. Test instances tend to become more difficult, if batch size restrictions are present compared to the case in which minimal and maximal lot size restrictions are present. The difficulty seems to be independent from the batch size, but different batch sizes have fairly different cost effects (Fig. 7-7). As expected, using the CLSPL as the basic model instead of the PLSP has been most advantageous, if setup costs are relatively cheap compared to inventory holding costs, while the effect has been negligible, if relatively high setup costs lead to CLSPL plans that almost match PLSP plans.

7.2.3 Computational Performance

All computational results in this section have been obtained using XpressMP release 2003G with standard settings as a MIP solver on a PC equipped with a Pentium IV processor with a clockspeed of 1.7 GHz and 256 MB of memory.

Comparison to Benchmark

The model formulation proposed in section 6.1.2 is first compared to the benchmark by Kallrath (1999).[348] The benchmark model formulation does not allow to relax the lot size restriction for the last lot within the planning horizon. Therefore, to allow for a fair comparison, a corresponding restriction is added to the new model formulation. If this additional restriction is present, this is indicated by "+K" in the second field of the description of the model formulation. Thus, both the Kallrath (1999) model formulation and the proposed model formulation have the same feasible region and the same optimal objective function value.

Fig. 7-9 shows the number of test instances solved to proven optimality over time for different model formulations. Apparently, solution behavior does not change by the addition of the additional constraint "+K" for the proposed model formulations. While the proposed model formulation solves a majority of test instances to proven optimality if minimum and maximum lot size restrictions are present, the Kallrath (1999) model formulation proves optimality for only 32 test

[348] See section 7.2.1. The model formulation can also be found in the appendix.

instances. If batch size restrictions are also included, the proposed model formulation still solves two thirds of all test instances of test set *TC*1 to proven optimality, while not one single test instance is solved to proven optimality by the Kallrath (1999) model formulation.

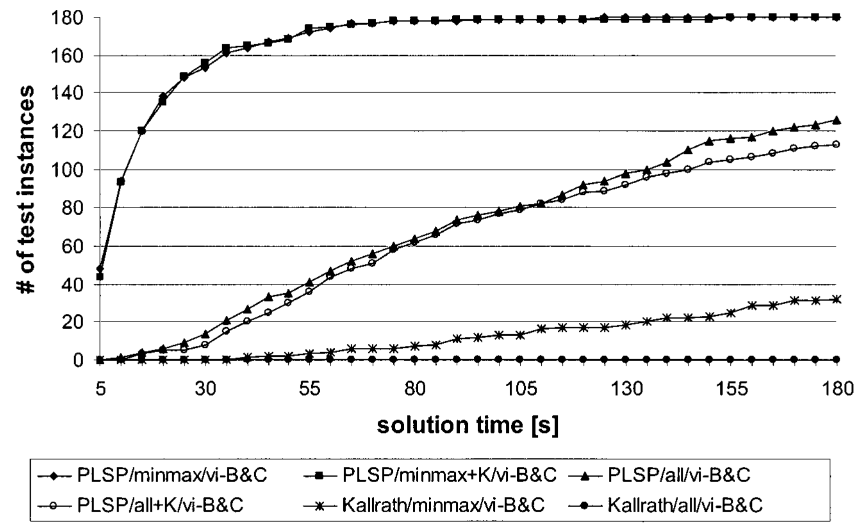


Fig. 7-9: Number of test instances solved to optimality over time (comparison to benchmark by Kallrath (1999), test set *TC*1).

Furthermore, as shown in Table 7-29, the Kallrath (1999) model formulation does not even provide feasible solutions to a considerable number of test instances within the time limit of 3 minutes. Also the time until the first feasible solution is found is considerably higher for the Kallrath (1999) model formulation than for the new model formulation.

Not only solution speed is much higher for the proposed model formulation compared to the benchmark by Kallrath (1999), but also the quality of solutions obtained is much better. Table 7-30 and Table 7-31 provide the remaining gaps after three minutes of computational time for model formulations with minimum and maximum lot size restrictions (Table 7-30) and for model formulations in which production has additionally to be in multiples of a batch size (Table 7-31). By comparing the gaps, one can observe that the remaining gap after three minutes of computational time is on average only 0.00% (1.31%) if minimum and maximum lot size (and batch size) restrictions are present. Thus, the solutions obtained by the new model formulation are almost optimal. The corresponding gaps for the Kallrath (1999) model formulation are 9.03% and 126.76%. The question whether these gaps are due to bad lower bounds or due to bad solutions is answered, if solutions for both model formulations (the new one and the one of Kallrath (1999)) are set in relation to the same lower bound – the optimal PLSP solutions. Solutions to the Kallrath (1999) model formulation are approximately 4% worse (more expensive) if minimal and maximal lot size restrictions are present, but 90% worse, if additionally batch size restrictions need to be applied. This means, a good plan (new model formulation) might be implemented at roughly 60%[349] of the cost of a bad plan (Kallrath (1999) model formulation). And this number only holds for the subset of test instances the Kallrath (1999) model formulation has been able to obtain a feasible plan at all.

Table 7-29: Solution speed of model formulations with different lot size restrictions for test sets *TC*1.

Number of test instances for which	PLSP/ * /vi-B&C		Kallrath/	PLSP/ * /vi-B&C		Kallrath/
* =	minmax	minmax+K	minmax/ vi-B&C	all	all+K	all/ vi-B&C
- a feasible solution is found within 15s	180	180	45	177	174	4
- a feasible solution is found within 30s	180	180	118	180	180	19
- a feasible solution is found within 60s	180	180	163	180	180	41
- a feasible solution is found within 90s	180	180	176	180	180	69
- a feasible solution is found within 180s	180	180	180	180	180	96
- optimality is proven within 180s	180	180	32	126	113	0
Average time (in seconds) to						
- find a first solution	2.72	2.81	28.73	6.12	6.25	37.99
- find an optimal solution[(a)]	11.29	11.47	90.34	61.70	60.14	-
- prove optimality[(a)]	14.97	15.24	112.28	84.56	81.89	-

[(a)] Only for test instances with proven optimality

Comparison of Different Variants

In section 6.1.2.1 it has been argued, that the lot size extension can either be modeled by defining variables Y_{jt} as binary variables or by using (6-4) as regular constraints. According to Fig. 7-10 this seems to have no impact on the solution behavior of different model formulations based on test set *TC*1. On the other hand, a difference is observed between the model formulation using valid inequalities only at the root node (C&B) and the one using the model cut feature of XpressMP (B&C). Using valid inequalities only at the root node seems to speed up the solution process at least in the beginning. At the time limit the model formulation using the C&B approach has found only seven optimal solutions more than the B&C approach (133 vs. 126), while after 90 seconds the advantage has peaked at 30 test instances (104 vs. 74).

349 From the last row in Table 7-31 one can obtain the average cost of solutions to model formulation PLSP/all+K/vi-B&C (135.04%) and Kallrath/all/vi-B&C (226.80%). Thus, the cost ratio of these two model formulations is about 60% (≈135.04/226.8).

Table 7-30: Gap (upper bound: best solution found within a time limit of 3 minutes; lower bound: bound after 3 minutes) and cost comparison to optimal PLSP solutions for PLSP/minmax/vi-B&C and benchmark (test set *TC*1).

	PLSP/minmax/ vi-B&C	PLSP/minmax+K/ vi-B&C	Kallrath/ min-max/vi-B&C
Gap (average)	0.00%	0.00%	9.03%
Gap (standard deviation)	0.00%	0.00%	7.68%
Gap (minimum)	0.00%	0.00%	0.00%
Gap (maximum)	0.00%	0.00%	35.66%
Average cost increase vs. PLSP	4.46%	4.79%	8.87%

Table 7-31: Gap (upper bound: best solution found within a time limit of 3 minutes; lower bound: bound after 3 minutes) and cost comparison to optimal PLSP solutions for PLSP/all/vi-B&C and benchmark (subset of test set *TC*1 for which the benchmark found a feasible solution within the time limit; 96 test instances).

	PLSP/all/ vi-B&C	PLSP/all+K/ vi-B&C	Kallrath/all/ vi-B&C
Gap (average)	1.75%	1.31%	126.76%
Gap (standard deviation)	3.01%	2.56%	64.51%
Gap (minimum)	0.00%	0.00%	24.24%
Gap (maximum)	13.48%	12.92%	331.92%
Average cost increase vs. PLSP	29.44%	35.04%	126.80%

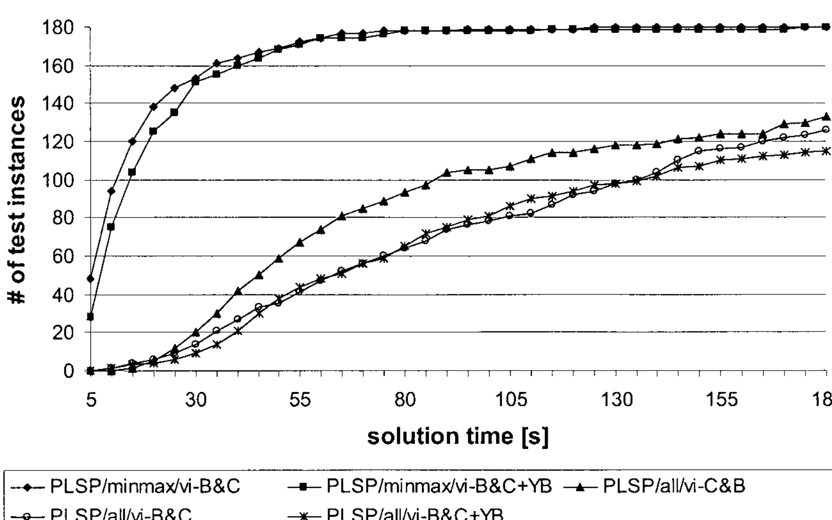


Fig. 7-10: Number of test instances solved to optimality over time (different model formulations, test set *TC*1).

Fig. 7-11 shows the speed up for the batch flow scenario if the preprocessing procedure outlined in section 6.1.2.1 is applied. Approximately half of the test instances seem to become fairly easy, just by rearranging demands. 90 out of 180 test instances are solved to proven optimality after at most 22 (31) seconds by model formulation PLSP/bat+flow/vi-B&C+pp (PLSP/all+flow/vi-B&C+pp). Thus, by applying the preprocessing procedure, the batch flow scenario, which has been identified as the most difficult in the beginning (Fig. 7-6), gets suddenly a lot easier.

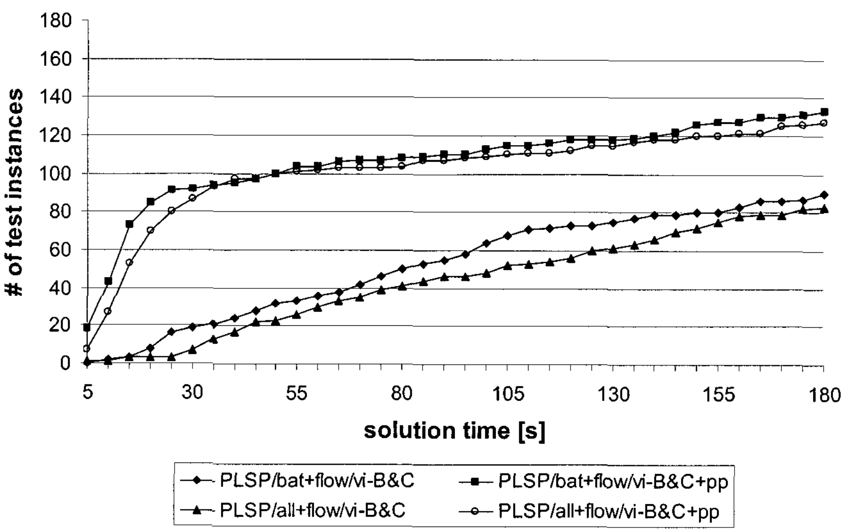


Fig. 7-11: Number of test instances solved to optimality over time (impact of preprocessing on batch flow scenario, test set *TC*1).

With Table 7-32 solution speed and solution quality can be analyzed a bit further. In the first two columns, the C&B-approach is compared to the model cut feature of XpressMP (B&C). Applying valid inequalities only at the root node (C&B) leads to much smaller gaps based on the linear relaxation after automatic cut generation (LB^{XLP}). On the other hand, this takes a lot of time. A first feasible solution has been obtained after 31 seconds on the average compared to just 6 seconds for the B&C approach. After three minutes of computation, both approaches are about even strength, proving optimality for nearly the same number of test instances (126 and 133) and finishing with roughly the same final gap (based on the lower bound after three minutes, LB^3).

Preprocessing in the batch flow scenario carries only minor computational effort. The time to find a first feasible solution roughly stays the same, whether preprocessing is applied or not, but integrality gaps are halved.

Two variants to allow for consecutive campaigns of the same product have been proposed in section 6.1.2.1. The first variant (CC1) was to simply double the number of variables (two for each product) and to switch between these two sets of variables, whenever a product needed to be set up twice or more. From Fig. 7-12 it is obvious that this variant is computationally more expensive than the second approach (denoted by CC2). This is especially true, if batch size restrictions need to be obeyed, because this doubles not only the number of binary, but also the number of integer variables of the model formulation. Moreover, variant CC2 performs almost as well as the basic variant, which does not allow two or more consecutive campaigns of the same product. So variant CC2 is clearly the best choice.

Table 7-32: Solution speed and solution quality of different model formulations for test sets *TC*1.

	PLSP/					
Number of test instances for which	all/ vi-B&C	all/ vi-C&B	bat+flow/ vi-B&C	bat+flow/ vi-B&C +pp	all+flow/ vi-B&C	all+flow/ vi-B&C +pp
- a feasible solution is found within 15s	177	3	163	154	163	160
- a feasible solution is found within 30s	180	87	174	171	174	174
- a feasible solution is found within 60s	180	179	177	178	176	178
- a feasible solution is found within 90s	180	180	179	178	176	178
- a feasible solution is found within 180s	180	180	179	180	179	179
- optimality is proven within 180s	126	133	90	133	83	126
Average time (in seconds) to						
- find a first solution	6.12	30.96	8.85	9.28	10.01	8.73
- find an optimal solution[(a)]	61.70	57.08	57.03	33.52	57.06	33.00
- prove optimality[(a)]	84.56	67.68	77.32	38.13	87.69	37.89
Gaps						
Average gap [%] (lower bound: LB^{XLP})	23.66	12.16	20.72	10.99	21.63	12.10
Average gap [%] (lower bound: LB^3)	1.75	1.66	5.00	1.84	5.43	2.10
Minimum gap [%] (lower bound: LB^3)	0.00	0.00	0.00	0.00	0.00	0.00
Maximum gap [%] (lower bound: LB^3)	13.48	19.41	23.14	21.83	29.38	16.83

[(a)] Only for test instances with proven optimality

Fig. 7-13 shows the number of test instances solved to optimality over time, if not the PLSP, but the CLSPL is chosen as the basic model. Surprisingly, not batch size restrictions seem to make the problem difficult, but minimal lot size restrictions. Even the number of test instances solved to proven optimality after three minutes for the version with all restrictions present (minimal, maximal and batch size restrictions) is greater than the number of test instances solved if only minimal and maximal lot size restrictions need to be respected.

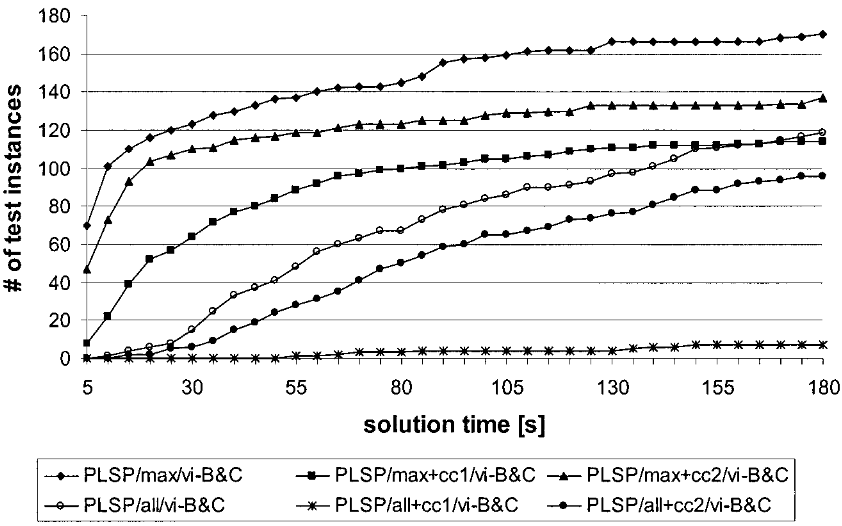


Fig. 7-12: Number of test instances solved to optimality over time (different model formulations with consecutive campaigns, test set *TC*1 and $maxlot_j$=1,500).

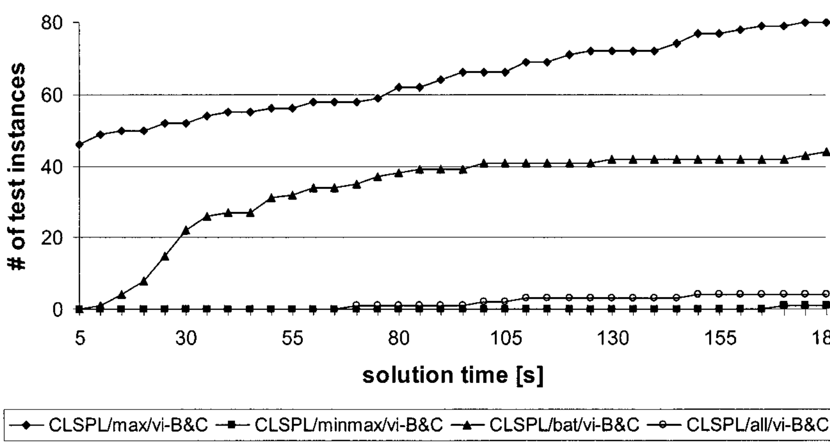


Fig. 7-13: Number of test instances solved to optimality over time (different lot size restrictions, test set *TC*1, basic model: CLSPL).

As test set *TC*1 (J=4, T=20) seems to be fairly easy – at least if the proposed modeling and solution approach is used – the number of products has been increased (J=6) to yield test set *TC*2, the number of periods has been increased

(T=40) to yield test set *TC*4 and both modifications yield test set *TC*5.[350] Table 7-33 compares, how well these test sets are solved with a model formulation for the basic PLSP. All test instances of test set *TC*1 have been solved to proven optimality within three minutes with a remaining gap after three minutes (lower bound: LB^3) of 0.00%. Test set *TC*2 seems to be somewhat more difficult. It takes on average slightly more than four seconds to obtain a first feasible solution. After three minutes of computational time all test instances are proven to be optimal and a gap of 5.44% remains on average. However, the tougher challenge is to double the number of periods. Although feasible solutions to all test instances are still found in a reasonable amount of time, only half of the test instances are proven optimal within the time limit of three minutes.

Table 7-33: Solution speed and solution quality of model formulations PLSP/–/vi-B&C for different test sets (upper bound: best solution after three minutes or first solution).

Number of test instances for which	*TC*1	*TC*2	*TC*4	*TC*5
- a feasible solution is found within 15s	180	180	133	10
- a feasible solution is found within 30s	180	180	180	68
- a feasible solution is found within 60s	180	180	180	161
- a feasible solution is found within 90s	180	180	180	180
- a feasible solution is found within 180s	180	180	180	180
- optimality is proven within 180s	180	180	95	70
Average time (in seconds) to				
- find a first solution	1.17	4.16	11.03	36.81
Average gap [%] (lower bound: LB^{XLP})	4.81	4.88	5.87	6.35
Average gap [%] (lower bound: LB^3)	0.00	0.00	3.05	4.57

Table 7-34: Solution speed and solution quality of model formulations PLSP/minmax/vi-B&C for different test sets (upper bound: best solution after three minutes or first solution).

Number of test instances for which	*TC*1	*TC*2	*TC*4	*TC*5
- a feasible solution is found within 15s	180	174	33	0
- a feasible solution is found within 30s	180	180	155	0
- a feasible solution is found within 60s	180	180	180	58
- a feasible solution is found within 90s	180	180	180	150
- a feasible solution is found within 180s	180	180	180	179
- optimality is proven within 180s	180	153	46	9
Average time (in seconds) to				
- find a first solution	2.72	8.19	20.68	71.36
Average gap [%] (lower bound: LB^{XLP})	7.27	8.77	9.38	14.54
Average gap [%] (lower bound: LB^3)	0.00	0.57	5.65	12.82
Average gap [%] (lower bound: $PLSP^3$)	4.46	6.80	4.41	10.37

Table 7-34 shows the same behavior for a model formulation which respects minimum and maximum lot size restrictions. Here, the gaps are also reported taking the PLSP solution after three minutes as a lower bound ($PLSP^3$). One can observe, that the cost of respecting these restrictions increases with the number of

350 See also section 7.2.1.

products. Solutions to test sets *TC*1 and *TC*4 (both *J*=4) show a lower cost increase than solutions to test sets *TC*2 and *TC*5 (both *J*=6). Moreover, solutions to test sets with a higher number of periods (*TC*4, *TC*5 vs. *TC*1, *TC*2) seem to have a higher cost associated with obeying minimal and maximal lot size restrictions, but this may well be due to the fact, that these solutions are not as good as those obtained for test sets *TC*1 and *TC*2 (see the remaining gap after three minutes).

The problem gets most difficult, if also batch size restrictions need to be obeyed (Table 7-35). Although feasible solutions have been obtained for each test instance, this took quite a while for some test instances. The maximum computational time to obtain a first feasible solution for test set *TC*2 (*TC*4, *TC*5) has been 229 (279, 1637) seconds. Interestingly, with respect to optimal PLSP solutions, here the cost increase is lower if more products need to be scheduled.

Table 7-35: Solution speed and solution quality of model formulations PLSP/all/vi-B&C for different test sets (upper bound: best solution after three minutes or first solution).

Number of test instances for which	*TC*1	*TC*2	*TC*4	*TC*5
- a feasible solution is found within 15s	177	33	0	0
- a feasible solution is found within 30s	180	138	101	0
- a feasible solution is found within 60s	180	172	160	2
- a feasible solution is found within 90s	180	175	170	40
- a feasible solution is found within 180s	180	178	178	139
- optimality is proven within 180s	126	4	0	0
Average time (in seconds) to				
- find a first solution	6.12	27.12	36.94	158.97
Average gap [%] (lower bound: LB^{XLP})	23.66	27.38	33.97	47.56
Average gap [%] (lower bound: LB^3)	1.75	16.27	27.73	44.85
Average gap [%] (lower bound: $PLSP^3$)	29.44	26.20	33.61	31.51

Decomposition Heuristic

In Fig. 7-14 and Fig. 7-15 the performance of the temporal decomposition heuristic is evaluated for the basic PLSP. Fig. 7-14 shows the minimum, maximum and average deviation from the PLSP solution found by model formulation PLSP/–/vi-B&C after three minutes for test sets *TC*1, *TC*4 and *TC*5. For each test set the solution quality is given for the model formulation with respect to a time limit (e.g., PLSP (30s)), one run of the decomposition heuristic (e.g., parameter combination 6/0/3 (15)) and a multiple run[351] of the decomposition heuristic (Best). In parenthesis, the actually used computational time is indicated.

Fig. 7-14 can be interpreted as follows. For test set *TC*5 the solution quality after 90 seconds for model formulation PLSP/–/vi-B&C ranges between 0.0% and 11.0% and averages 1.8% (indicated by a mark ♦ in Fig. 7-14). On the other hand,

[351] A multiple run of the decomposition heuristic means to run the heuristic with five different parameter combinations. The computational times of these runs are added, while the best solution obtained by any of these runs is chosen as the solution of the multiple run. Unless otherwise noted, the parameter combinations for the multiple run have been 4/0/2, 5/0/2, 6/0/2, 5/0/3, 6/0/3.

solutions of the temporal decomposition heuristic with parameter combination 6/0/3 lie between –9.8% and 4.5% averaging –3.0%. This means, the temporal decomposition heuristic has been able to find solutions, which are on average 3.0% less expensive than those found by model formulation PLSP/–/vi-B&C after three minutes. Furthermore, the heuristic needed only approximately one tenth of the time (22.1s) to do so. The multiple run produced only slightly better results, providing solutions which are on average 3.6% less expensive than the reference values.

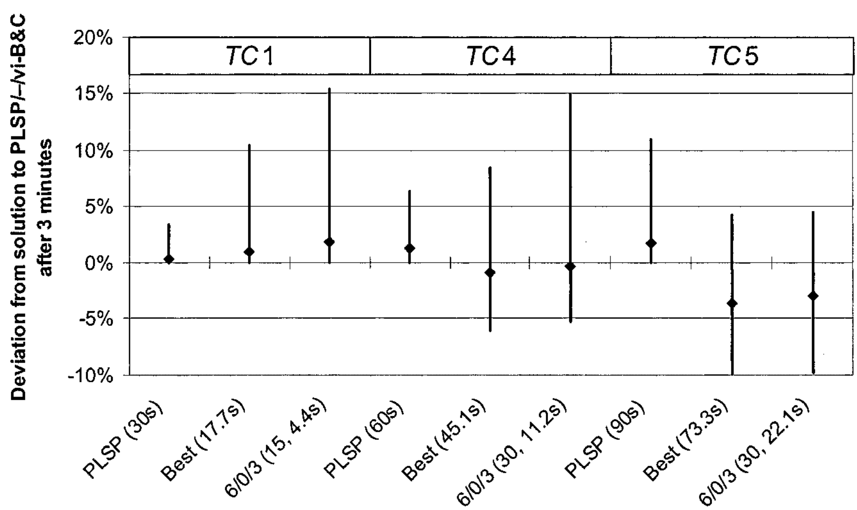


Fig. 7-14: Comparison of solution quality of model formulation PLSP/–/vi-B&C and the temporal decomposition heuristic for test sets *TC*1, *TC*4 and *TC*5.

Fig. 7-15 provides a map of solution quality and solution speed of the temporal decomposition heuristic for test set *TC*4. Again, the solutions to model formulation PLSP/–/vi-B&C after three minutes are taken as reference values. This is indicated by the point “PLSP (180s)” on the map. The solutions of model formulation PLSP/–/vi-B&C after different amounts of computational time mark the frontier the heuristic has to cross, if it wants to add value. Points of the heuristic must lie either left of this frontier or below it. A point left of the frontier means, that the heuristic is able to provide the same solution quality in less time. A point below this frontier indicates, that the heuristic provides better solutions in the same amount of time.

Apparently, one run of the decomposition heuristic takes only a short amount of time. The solution is obtained on average within 7.0 seconds for parameter combination 4/0/2 and within 11.1 seconds for parameter combination 6/0/3. The latter even shows a slightly better solution quality than model formulation PLSP/–/vi-B&C after three minutes. Moreover, solution quality can be improved by performing a multiple run. However, the cost associated with this is additional computational effort.

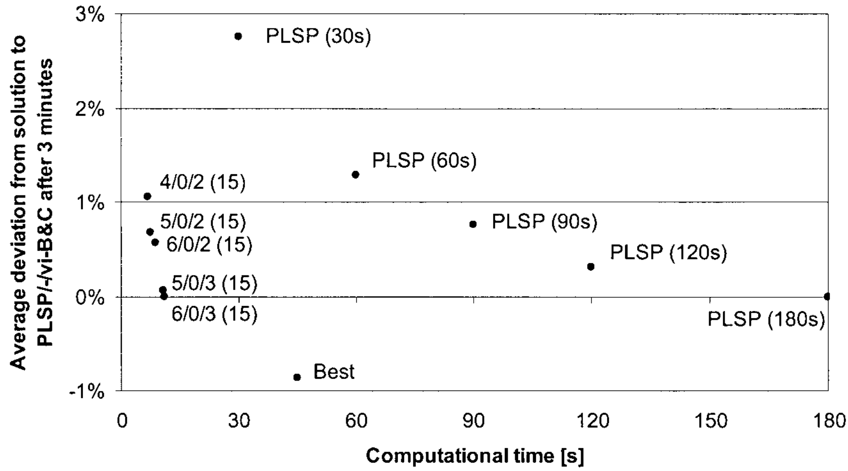


Fig. 7-15: Comparison of solution quality and speed of model formulation PLSP/–/vi-B&C and the temporal decomposition heuristic for test set *TC*4.

Table 7-36: Average gap between best solution and the best solution found by PLSP/–/vi-B&C after 3 minutes for test sets *TC*4 and *TC*5 ("Best" refers to a multiple run of the heuristic).

*TC*4	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/–/vi-B&C (60s)	0.8	1.1	2.0	1.1	1.3	1.4	1.3	1.2	1.6	1.0
5/0/2 (15, 7.4s)	0.9	0.2	1.0	1.1	1.0	-0.1	-0.2	1.5	1.1	0.3
Best (45.1s)	-0.7	-0.9	-1.1	-0.2	-0.3	-2.1	-1.4	-0.3	-0.7	-1.0

*TC*5	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/–/vi-B&C (90s)	1.2	1.7	2.4	1.3	1.8	2.1	1.5	2.1	2.1	1.4
6/0/3 (30, 22.1s)	-2.3	-2.5	-4.0	-2.8	-3.4	-2.8	-3.6	-2.3	-2.8	-3.1
Best (73.3s)	-3.1	-3.6	-4.1	-3.5	-4.4	-2.9	-4.2	-3.0	-3.5	-3.7

In Table 7-36 the performance of the heuristic is evaluated for different test set characteristics. Different levels of capacity utilization seem to have only a minor impact on the performance of the heuristic for test set *TC*4, while in test set *TC*5 the solution quality of the heuristic is better for higher capacity utilization levels. On the other hand, in test set *TC*5 the TBO profile seems to have only a minor impact on the performance of the heuristic, while the solution quality of the heuristic is best for TBO profile "high" in test set *TC*4. Thus, the heuristic performs best in those cases, that have been identified to be the most difficult (high capacity utilization and relatively high setup costs; Table 7-25). Moreover, the heuristic performs better the more stable the demand situation is, while the length of setup times has only minor impact.

In Fig. 7-16 the solution map which has been provided for the basic PLSP in Fig. 7-15 is drawn for the PLSP with additional restrictions on minimal and maximal lot sizes. Note, that the scale of the ordinate has changed, but the shape of the frontier has not. Here, all parameter combinations tested for the heuristic provide the same or a better solution quality than model formulation PLSP/minmax/vi-B&C after three minutes, which has been used as a basis here "PLSP (180s)". The best parameter combination has been 6/0/3 using on average 20.6 seconds to obtain solutions that are on average 1.4% less expensive than the reference values. Again, solution quality can be improved by performing a multiple run, which takes 82.6 seconds on the average and provides solutions that are on average 2.1% less expensive than the reference values.

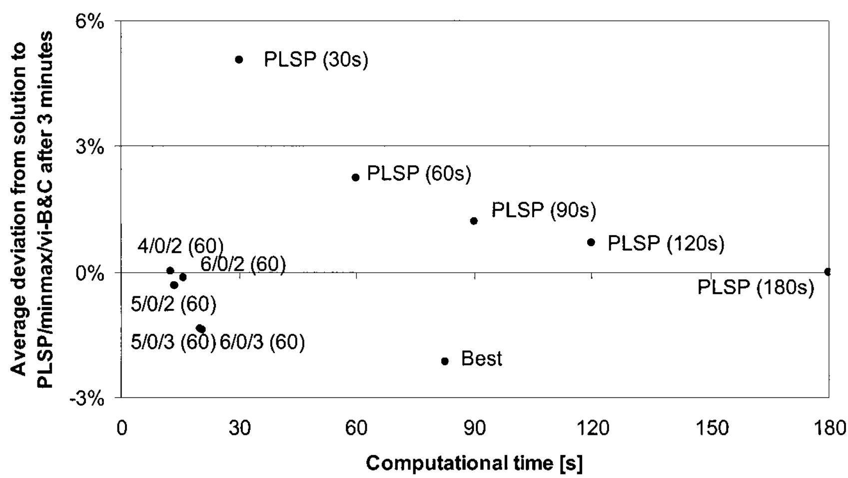


Fig. 7-16: Comparison of solution quality and solution speed of model formulation PLSP/minmax/vi-B&C and the temporal decomposition heuristic for test set *TC*4.

The analysis of solution quality with respect to test instance characteristics in Table 7-37 reveals no new insights compared to the results for the plain PLSP (Table 7-36). Again, the heuristic performs relatively best if capacity utilization levels are high and demand variation is low.

To evaluate the absolute solution quality of the heuristic, a subset consisting of ten randomly chosen test instances has been tried to solve to optimality on a much faster PC (Pentium IV, 2.4 GHz, 256 MB) with model formulation PLSP/minmax/vi-B&C. A time limit of 12 hours has been imposed on these runs. Three test instances have been proven optimal, while the average final gap after 12 hours has been 3.75%. The multiple run of the heuristic provides solutions, which are on average only 1.35% worse (worst case: 5.57%). However, the computational effort of the multiple run is only slightly more than one minute on a much slower PC.

Table 7-37: Average gap between best solution and the best solution found by PLSP/minmax/vi-B&C after 3 minutes for test set *TC*4 ("Best" refers to a multiple run of the heuristic with parameter combinations 4/0/2, 5/0/2, 6/0/2, 5/0/3 and 6/0/3).

	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/minmax/vi-B&C (90s)	0.9	1.3	1.4	1.7	0.9	1.1	1.3	1.1	1.1	1.3
6/0/3 (60, 20.6s)	-0.5	-1.4	-2.2	-1.5	-0.4	-2.1	-1.9	-0.8	-1.3	-1.5
Best (82.6s)	-1.2	-2.1	-3.0	-2.5	-1.3	-2.6	-2.5	-1.8	-2.1	-2.2

Finally, the decomposition heuristic has been evaluated for model formulations, which also have to respect batch size restrictions. It has been pointed out in section 6.2.2 that the decomposition heuristic needs to be altered to take also batching decisions in periods preceding the rolling window into account. Three variants to do this have been proposed (*var*, *fix* and *fix1*). Table 7-38 evaluates their behavior for test set *TC*4. Each individual parameter combination does not find feasible solutions to all test instances. Therefore, multiple runs should be performed. Variant *var* finds the most feasible solutions for each individual parameter combination, but is also the slowest rolling scheme as has been expected. On the other hand, variant *fix1* finds much more feasible solutions than variant *fix* without using too much computational time and without deteriorating the solution quality. Thus, *fix1* should be preferred over *fix*. The best results have been obtained performing a multiple run combining *fix* and *fix1*.

Table 7-38: Evaluation of different rolling schemes for the decomposition heuristic, if batch size restrictions are present. Gap refers to the best solution found in relation to the best solution found by PLSP/all/vi-B&C after three minutes. "Best" refers to a multiple run of the two parameter combinations presented.

	Gap			Time [s]	# feasible
	Average	Min	Max		solutions
PLSP/all/vi-B&C (60s)	3.36	0.00	22.35	60.00	160
5/0/2 (*fix*, 30)	8.18	-4.18	41.66	14.90	148
5/0/3 (*fix*, 30)	7.23	-6.07	28.54	22.58	147
Best of *fix*	5.61	-6.07	22.42	34.27	168
5/0/2 (*fix1*, 30)	8.57	-5.71	37.15	17.16	170
5/0/3 (*fix1*, 30)	7.72	-7.88	29.53	25.24	162
Best of *fix1*	5.84	-7.88	22.42	42.40	178
Best of *fix* and *fix1*	4.06	-7.88	22.42	76.67	178
5/0/2 (*var*, 30)	9.05	-4.43	43.58	65.58	176
5/0/3 (*var*, 30)	10.36	-6.08	51.63	95.85	179
Best of *var*	6.75	-6.08	24.08	160.90	179

For several instances (e.g., 21 for "Best of *fix*"), the decomposition heuristic finds better solutions than model formulation PLSP/all/vi-B&C (Table 7-38). Table 7-39 aims at identifying, in which cases the heuristic is expected to result in better solutions and in which cases the complete model formulation should be used. One individual parameter combination and a multiple run are compared to model formulation PLSP/all/vi-B&C.

The heuristic tends to perform better, if capacity utilization is high. One possible explanation for this phenomenon is, that when capacity utilization is low demands are balanced over longer periods to build lots. Thus, the decomposition heuristic may suffer from some kind of planning horizon effect here. Furthermore, the heuristic tends to perform better, if demand variation is low, which points into the same direction. Hence, when improving the heuristic, effort should be devoted to reducing these planning horizon effects.

Table 7-39: Average gap between best solution and the best solution found by PLSP/all/vi-B&C after 3 minutes for test set *TC*4.

	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/all/vi-B&C (30s)	5.7	7.1	10.0	8.3	6.2	1.9	7.4	7.0	7.8	6.6
5/0/3 (*fix*, 30, 22.6s)	8.3	6.1	7.3	9.4	6.9	4.6	6.5	8.0	7.4	7.1
PLSP/all/vi-B&C (60s)	2.4	3.1	5.1	3.0	3.5	3.6	3.9	2.8	3.7	3.0
Best of *fix* and *fix1* (76.7s)	4.0	4.1	4.1	5.4	4.1	2.7	3.4	4.7	3.7	4.5

Summing up, computational testing reveals that the proposed model formulations clearly outperform the benchmark by Kallrath (1999). Better solutions as well as better lower bounds have been computed with the new model formulation in less time. The preprocessing procedure outlined for the batch flow scenario has been shown to increase computational performance severely. Moreover, if maximal lot size restrictions are present, the second variant to model consecutive lots of the same product has proven to be the more viable approach. With respect to different test sets, an increase in the number of products consumed less additional computational effort than an increase in the number of periods. However, for these test sets, the temporal decomposition heuristic provided good results with reasonable computational effort.

7.3 Time Continuity – Setup Operations

7.3.1 Test Sets and Benchmark

The focus of this section will be on period overlapping setup times. As this scenario has – so far – not been addressed computationally in literature, a new test set needs to be defined. This new test set comprises two classes of test instances *TS*1 and *TS*2. Furthermore, test set *TC*1[352] will be used here again.

Test sets *TS*1 and *TS*2 both comprise 40 test instances with J=3 products and a planning horizon of T=15 and T=30 periods. Capacity (c_t) is set to 100 for each period in each test instance. Production coefficients (a_j) are set equal to 1 for each product, while inventory holding cost coefficients (h_j) are set to 1, 2 and 3 for products 1, 2 and 3. Setup costs (sc_j) are proportional to inventory holding costs and are set to 400, 800 and 1200 for products 1, 2 and 3. Two capacity situations will be evaluated. In the first one (tight capacity) the demand per period for each product is equally distributed between 10 and 30, whereas in the second situation (loose capacity) less customer demand is assumed (equally distributed between 8 and 24). Five different demand series have been randomly generated for each situation. To make test instances feasible, demand has been set to 0 for all products in periods 1–3 (test instances 1–10, 16–30, 36–40), 1–4 (11–15, 31–35), 1–5 (41–50, 56–80), 1–6 (51–55).

Four different setup time profiles have been generated. In the first setup time profile, setup times for all products (st_j) are equal to 40. This means a setup operation consumes 40% of available period capacity. In the second and third scenario, setup times are even higher (80 and 120). In the latter case, it is impossible to solve the problem with a standard lot sizing model formulation, because no setup operation fits completely into one planning period. A fourth setup profile mixes the different setup times. Here, setup times are 40, 80 and 120 for products 1, 2 and 3. Table 7-40 shows the generation scheme for test sets *TS*1 and *TS*2.

Table 7-40: Test instance generation scheme for test sets *TS*1 and *TS*2.

		Test instances							
TS1	T=15	1–5	6–10	11–15	16–20	21–25	26–30	31–35	36–40
TS2	T=30	41–45	46–50	51–55	56–60	61–65	66–70	71–75	76–80
Setup	j=1	40	80	120	40	40	80	120	40
times	j=2	40	80	120	80	40	80	120	80
st_j	j=3	40	80	120	120	40	80	120	120
capacity		tight	tight	tight	tight	loose	loose	loose	loose

In section 3.4 it has been shown that a similar model formulation known from literature is not correct. Therefore, this model formulation[353] can only be used as a benchmark to some extent. Instead, the basic models PLSP and CLSPL are used to serve as a benchmark.

7.3.2 Analysis of Solutions

To analyze the behavior of different model formulations, the following notation is introduced to distinguish between the model formulations (Table 7-41). The

352 See section 7.2.1.

353 See section 3.4 and e.g. Drexl and Haase (1995) pp. 81-82.

first field represents the basic model formulation, which is either the PLSP or the CLSPL. The second field characterizes the different variants to model period overlapping setup times. Finally, the third field describes the use of valid inequalities. These are either used ("vi") or not ("–"). If valid inequalities are used, they can be applied at the root node only (cut-and-branch approach, "C&B") or throughout the branch-and-bound search. The latter is indicated by "B&C". Again, the model cut feature of XpressMP is used here. E.g., the abbreviation "PLSP/POST1/–" describes a model formulation using the PLSP as the basic model formulation and variant POST1 as described in section 6.1.3.1 to model period overlapping setup times. Valid inequalities are not used in this example.

Table 7-41: Notation of different model formulations (POST).

Parameter	Value	Comment
1	PLSP	Basic model formulation *PLSP* (I&L)
	CLSPL	Basic model formulation *CLSPL* (I&L)
2	POST1	*P*eriod *O*verlapping *S*etup *T*imes Variant *1* (see section 6.1.3.1 [PLSP] and 6.1.3.2 [CLSPL])
	POST2	*P*eriod *O*verlapping *S*etup *T*imes Variant *2* (see section 6.1.3.1)
3	–	No valid inequalities
	vi-C&B	Valid inequalities (see section 6.1.3.1), *C*ut&*B*ranch approach (=valid inequalities only applied in the root node, see section 5.2.1.2)
	vi-B&C	Valid inequalities (see section 6.1.3.1), *B*ranch&*C*ut approach (=model cut feature of XpressMP, see section 5.2.1.4)

In Table 7-42 the optimal solutions of test set *TS*1 are compared. The model with period overlapping setup times is benchmarked with the basic PLSP[354] and Haase's (1994)[355] model formulation. The PLSP is solved to proven optimality the fastest, needing approximately a third of the computational effort compared to PLSP/ POST2/vi-C&B. On the other hand, the integrality gap (upper bound: optimal solution; lower bound: LB^{XLP}) of the model formulation allowing for period overlapping setup operations is much lower compared to the standard PLSP. Furthermore, the standard PLSP will find only feasible solutions to 19 out of 40 test instances, because it cannot find any feasible solutions for those 20 instances in which at least one setup time exceeds the period length. In 6 out of these 19 test instances the optimal solution of the PLSP matches the one of the model allowing for period overlapping setup operations. All other optimal solutions to the basic PLSP are inferior, because the plan needs to fit setup times completely into one period. This inferiority (cost increase) increases with the length of setup times present in the planning problem (test instances 6–10 vs. 1–5, 26–30 vs. 21–25) as well as with the capacity utilization (test instances 1–10 vs. 21–30).

The second benchmark (Haase's (1994) model formulation) needs on average three times the computational effort compared to PLSP/POST2/vi-C&B. Furthermore, its integrality gaps are higher. It finds the correct optimal solution only in 14 of 40 test instances, but due to a modeling mistake fails in the remaining 26 test

354 See sections 2.3.3 and 6.1.1.1.

355 Cf. Haase (1994) pp. 31-35 and Drexl and Haase (1995) pp. 81-82.

instances. Therefore it is not deemed appropriate as a benchmark and is discarded from further analyses.

Table 7-42: Comparison of solutions for test set *TS*1.

Test instances	PLSP/POST2/ vi-C&B Avg. time [s]	Gap (LB^{XLP})	PLSP Avg. time [s]	Gap (LB^{XLP})	Same solution (POST)	Cost increase (POST)	Haase Avg. time [s]	Gap (LB^{XLP})	Same solution (POST)	Failed
1–5	7.6	12.4%	2.0	21.3%	2	0.5%	8.4	22.7%	5	0
6–10	35.6	29.1%	12.6	53.7%	0	8.4%	127.2	35.4%	0	5
11–15	26.6	22.9%	-	-	-	-	43.6	35.5%	2	3
16–20	30.0	28.0%	-	-	-	-	128.2	38.3%	0	5
21–25	6.0	8.7%	1.2	15.8%	4	0.1%	4.2	17.8%	5	0
26–30	18.0	22.1%	9.4	39.3%	0	5.0%	36.8	27.4%	0	5
31–35	12.0	17.2%	-	-	-	-	27.4	29.3%	2	3
36–40	24.4	24.2%	-	-	-	-	56.6	31.3%	0	5

Table 7-43 analyzes the difficulty of different test instance characteristics for test set *TS*1. Average time to prove optimality (time limit: 5 minutes) and integrality gaps are reported. With respect to the capacity situation, tight capacities seem to make the problem more difficult. Both, integrality gaps and average computational times are higher in this case. Regarding the different setup time profiles, the easiest scenario seems to be the one with rather small setup times. Surprisingly, test instances with setup times exceeding a period's capacity (120) seem to be easier than those with setup times somewhat below a period's capacity (80). Finally, the setup time profile with mixed setup times seems to have the same difficulty like its mean setup time suggests.

The observations do not change whether different basic models are chosen (PLSP or CLSPL) or whether different modeling approaches are taken (POST1 or POST2). Of course, those two model formulations with the PLSP as the basic model are solved somewhat faster.

Table 7-43: Comparison of difficulty of test instances in test set *TS*1.

		PLSP/POST1/vi-B&C Avg. time [s]	Gap (LB^{XLP})	PLSP/POST2/vi-B&C Avg. time [s]	Gap (LB^{XLP})	CLSPL/POST1/vi-B&C Avg. time [s]	Gap (LB^{XLP})
Capacity	tight	17.8	29.1%	21.3	32.3%	76.9	33.0%
	loose	12.4	23.2%	13.1	25.6%	31.3	26.4%
Setup times	40	5.6	17.0%	3.8	16.9%	6.3	20.2%
	80	31.2	36.4%	25.4	36.0%	82.0	31.4%
	120	6.4	19.5%	14.4	28.4%	35.5	32.4%
	40, 80, 120	17.2	31.9%	25.1	34.4%	92.4	34.8%

Although test set *TC*1 contains rather short setup times (average length of one product's setup time in relation to a period's capacity: 5.27%), it is used here to gain additional insight in which cases the explicit modeling of period overlapping setup times is most beneficial. Again, the information on the generation scheme for test set *TC*1 as explained in section 7.2.2 is used.

In Table 7-44 the percentage deviation from optimal PLSP solutions is shown. Positive numbers indicate a cost decrease, while negative numbers indicate a cost increase compared to optimal PLSP solutions. Negative numbers are only possible, because not all test instances have been solved to optimality.[356]

Comparing the variants based on the PLSP as a basic model, there is not much to gain here. This has been expected, because the setup times are too small compared to a period's capacity to have a great impact if splitting them over two periods is allowed. Still minor improvements to optimal PLSP solutions are possible even for some test instances with model formulation PLSP/POST1/vi-B&C although this is never reflected in the averages reported here (which are all negative or zero). Much more can be gained, if more than one setup operation is allowed in each period. If the CLSPL is used as a basic model, massive cost reductions (on average 23.56%) can be achieved. These are higher for low capacity utilization, low demand variation and rather short setup times. The biggest impact has the TBO profile. In TBO profile "low" setup costs are low compared to inventory holding costs. This means setup operations will be performed rather frequently. In this case, the CLSPL plays at its strength, because it allows for more than one setup operation in each period in contrast to the PLSP.

Table 7-44: Average percentage deviation from optimal PLSP solutions of different model formulations (time limit: 5 minutes) for test set *TC*1.

	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/POST1/vi-B&C	-0.01	-0.06	-0.15	0.00	0.00	-0.21	-0.10	-0.05	-0.07	-0.08
PLSP/POST2/vi-B&C	0.00	0.00	0.02	0.00	0.00	0.02	0.00	0.01	0.01	0.01
CLSPL	24.84	23.89	21.95	55.95	12.74	1.98	23.95	23.16	23.47	23.64
CLSPL/POST1/vi-B&C	24.46	23.48	21.32	55.95	12.34	0.98	23.54	22.64	23.12	23.05

Summarizing the results, it is worth modeling a period overlap of setup operations, if the ratio of the length of an (average) setup operation to the length of a period approaches one. Cost savings increase with this ratio, but also does computational effort to find optimal solutions. If setup operations exceed a period's capacity, the resultant problem tends to become easier to solve (based on the average gap between the optimal solution and the linear relaxation after automatic cut generation LB^{XLP}).

7.3.3 Computational Performance

All computational results in this section have been obtained using XpressMP release 2003G with standard settings as a MIP solver on a PC equipped with a Pentium IV processor with a clockspeed of 1.7 GHz and 256 MB of memory.

---

[356] A time limit of 5 minutes per test instance has been imposed on all model formulations. Cf. Table 7-45 for solution statistics.

Table 7-45 and Table 7-46 compare the computational effort of different model formulations with period overlapping setup times to the basic models PLSP and CLSPL for test set *TC*1. Optimality is proven for all test instances for the basic model formulation PLSP within the time limit of 5 minutes. With respect to the different variants to model period overlapping setup times (POST1 vs. POST2), POST2 consistently outperforms POST1 based on the number of test instances solved to proven optimality within the time limit (Table 7-45) and based on the average integrality gap (Table 7-46).

Fewer instances are solved to proven optimality with the CLSPL as a basic model, but if the CLSPL is used as a basic model the solution speed strongly depends on the characteristics of the test instances. The strongest influence stems from the TBO profile. If setup costs are low compared to inventory holding costs (TBO profile: low), integrality gaps are much lower for the CLSPL than for the PLSP, while the opposite is true, if setup costs are high compared to inventory holding costs (TBO profile: high). Furthermore, high capacity utilization seems to make the problem more difficult, while demand variation and absolute setup time length seem to have only minor effects here.

Table 7-45: Number of test instances (test set *TC*1, total: 180) solved to proven optimality within a time limit of 5 minutes.

	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP	60	60	60	60	60	60	90	90	90	90
PLSP/POST1/vi-B&C	56	51	43	60	58	32	67	83	78	72
PLSP/POST2/vi-B&C	60	59	57	60	60	56	86	90	89	87
CLSPL	30	27	19	59	16	1	33	43	37	39
CLSPL/POST1/vi-B&C	22	22	19	59	4	0	30	33	32	31

Table 7-46: Average gap [%] (upper bound: best solution found within 5 minutes; lower bound: LB^{XLP}) for test instances of test set *TC*1.

	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP	15.9	18.1	24.2	16.5	16.3	25.3	20.1	18.7	20.2	18.5
PLSP/POST1/vi-B&C	15.4	17.7	23.9	16.8	15.3	24.8	19.6	18.3	19.8	18.2
PLSP/POST2/vi-B&C	15.4	17.5	23.5	16.8	15.2	24.4	19.4	18.2	19.6	18.0
CLSPL	17.8	20.4	28.3	3.6	24.7	38.2	22.4	21.9	22.5	21.9
CLSPL/POST1/vi-B&C	18.4	20.9	29.4	3.7	25.2	39.7	23.1	22.7	23.1	22.7

Fig. 7-17 shows the number of test instances solved to proven optimality over time for different model formulations. It is obvious, that the basic PLSP is the easiest model formulation to be solved here. Variants POST2 and POST1 using the PLSP as a basic model show a similar behavior over time. Because a subset of test set *TC*1 (TBO profile: low) is solved much faster with the CLSPL as a basic model, these lines lie on top at the beginning, but are subsequently intersected by those lines representing model formulations with the PLSP as a basic model.

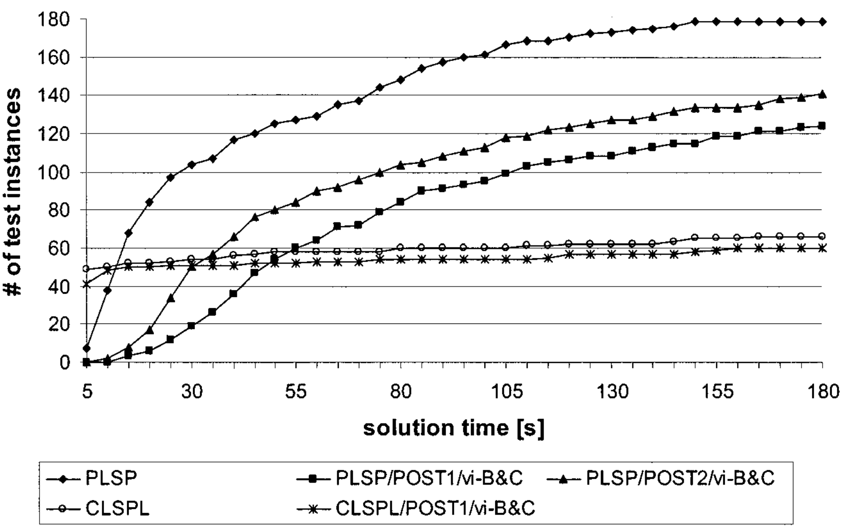


Fig. 7-17: Number of test instances solved to optimality over time (different model formulations, test set *TC*1).

Test set *TS*2 comprises test instances that cannot be solved to proven optimality within the time limit of 5 minutes. Again, the basic PLSP is only able to produce a feasible solution to 20 out of 40 test instances, because there is at least one setup time exceeding the period length present in each of the other 20 test instances. Compared to either PLSP/POST1/vi-B&C or PLSP/POST2/vi-B&C the basic PLSP model formulation has found a better solution ten times, the same solution once and a worse solution nine times. Still, the solution found with either of the model formulations allowing for period overlapping setup operations is on average 0.23% better (cheaper) than the one found by the basic PLSP model formulation.

Finally, Table 7-47 shows the impact of valid inequalities on the two variants to model period overlapping setup times (POST1 and POST2) based on test set *TS*2. The gaps are based on the lower bound after automatic cut generation (LB^{XLP}) to show the impact of valid inequalities, on the bound obtained after 5 minutes of branch-and-bound search (LB^{5}) to show the convergence over time as well as on the best bound obtained by any model formulation (LB^{BEST5}) to compare the solution quality based on one common lower bound for all model formulations.

Regarding variant POST1 the C&B approach provides lower initial gaps as well as lower gaps after 5 minutes of computational time. Still, the B&C approach provides better solutions. The variant without any valid inequalities is clearly outperformed by the two model formulations making use of valid inequalities. These observations hold also true for variant POST2. Comparing the two variants, POST2 seems to have a slight advantage over POST1, at least if valid inequalities are used. Only within the subset of test set *TS*2 with long setup times (120), POST1 outperforms POST2 with respect to providing better bounds as well as better solutions.

Table 7-47: Average gap [%] (upper bound: best solution found within a time limit of 5 minutes) to different lower bounds (XLP = bound after automatic cut generation; 5 = bound after 5 minutes; BEST5 = best bound (of all model formulations) after 5 minutes) for test set *TS2* for different model formulations.

	PLSP/POST1/–			PLSP/POST1/vi-B&C			PLSP/POST1/vi-C&B		
	LB^{XLP}	LB^{5}	LB^{BEST5}	LB^{XLP}	LB^{5}	LB^{BEST5}	LB^{XLP}	LB^{5}	LB^{BEST5}
40	253.2	68.6	17.1	38.7	13.9	11.7	27.8	14.2	12.5
80	294.4	88.1	26.2	52.5	26.7	21.8	38.7	24.4	22.7
120	322.4	97.4	26.1	49.6	20.8	18.5	30.1	17.3	17.0
40, 80, 120	288.3	81.0	24.4	50.0	21.1	19.1	36.8	20.8	20.6
	PLSP/POST2/–			**PLSP/POST2/vi-B&C**			**PLSP/POST2/vi-C&B**		
	LB^{XLP}	LB^{5}	LB^{BEST5}	LB^{XLP}	LB^{5}	LB^{BEST5}	LB^{XLP}	LB^{5}	LB^{BEST5}
40	289.6	64.4	14.5	38.7	12.5	9.8	26.4	12.1	9.8
80	331.6	83.9	25.6	51.7	24.0	20.4	35.4	21.5	20.0
120	369.1	98.5	26.5	70.6	42.4	23.5	55.7	43.3	27.8
40, 80, 120	318.5	77.3	22.7	52.8	25.4	19.9	38.7	24.9	20.6

Based on these results, variant POST2 to model period overlapping setup times seems to be slightly more favorable computationally compared to POST1. Still, this observation is not true for all setup time profiles and – even more important – variant POST2 might need to be altered, if the data changes. As has been argued above,[357] additional constraints apply if the assumptions on sp_j are not met. Model formulations using the CLSPL as a basic model are fairly easy to solve, if setup costs are low compared to inventory holding costs (TBO profile: low), but they are computationally much more expensive compared to model formulations using the PLSP as a basic model in all other cases.

7.4 Time Continuity – Resource Utilization

7.4.1 Test Set

Test set *TC*1 with additional restrictions on lot sizes, as it has been proposed in section 7.2.1, will be used throughout this section again. To analyze the effects different preconditions on resource utilization have, three resource profiles, forcing resources to be used to at least 55% (70%; 85%) are defined and their effects are computationally explored.

Model formulations, in which resources are modeled as if they are not allowed to be switched off, are indicated by "+On(55)", while the number in parenthesis indicates the required minimal resource utilization level. Regarding resources with off times, two variants have been proposed in section 6.1.4.1. "+Off1(55)" indi-

[357] See section 6.1.3.1.

cates that constraints (6-92) have been used in the model formulation, while "+Off2(55)" indicate the use of constraints (6-93) and (6-94).

7.4.2 Analysis of Solutions

The average cost increase of different restrictions on resource utilization compared to optimal PLSP solutions without any such restrictions is shown in Table 7-48. If resources are not allowed to have off times, too much product is produced resulting in much higher inventory holding costs. As argued in section 6.1.4.2, this scenario needs some further treatment (e.g., penalty costs for excess production, such that only the "correct" product is overproduced). Clearly, less excess production is necessary, if capacity utilization is high anyway. Moreover, the cost increase compared to optimal PLSP solutions is smaller, if inventory costs are relatively cheap (TBO profile: "high"). Variations in demand seem to have no influence, while shorter setup times have a much smaller cost effect than long setup times.

If resources are allowed to be switched off, a cost increase compared to optimal PLSP solutions without any restrictions on resource utilization still exists, but is almost negligible. Again, lower capacity utilization by demands leads to a higher cost increase, while the pattern changes compared to resources without off times for different TBO profiles. Here, the cost increase is bigger, if inventory costs are rather low.

Table 7-48: Average cost increase [%] (upper bound: best solution found within 3 minutes; lower bound: optimal PLSP solution without any restrictions on utilization) for test instances of test set *TC*1.

	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/–/vi-B&C	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
+On(55)	21.1	10.0	4.5	25.1	8.3	2.2	11.8	12.0	7.4	16.4
+On(70)	111.6	46.2	15.2	132.3	34.8	5.9	58.7	56.6	28.7	86.6
+On(85)	286.6	158.9	64.7	357.4	126.1	26.7	170.4	169.7	113.5	226.7
+Off1(55)	0.4	0.2	0.0	0.0	0.1	0.5	0.2	0.3	0.2	0.3
+Off1(70)	0.8	0.4	0.1	0.0	0.1	1.1	0.3	0.5	0.3	0.5
+Off1(85)	1.2	0.6	0.1	0.0	0.1	1.8	0.6	0.7	0.5	0.8

Table 7-49 and Table 7-50 show gaps compared to optimal solutions without any restrictions on resource utilization, if additionally minimal and maximal restrictions on lot sizes are present or batch size restrictions are applied.[358] The same patterns that have been identified for the basic PLSP model formulation (Table 7-48) are present here: Low capacity utilization (by demands) leads to higher cost increases, while the results differ between resources with and without off times for different TBO profiles.

[358] See Table 7-22 for the notation of model formulations.

Table 7-49: Average cost increase [%] (upper bound: best solution found within 3 minutes; lower bound: optimal solution to PLSP/minmax/vi-B&C without any restrictions on utilization) for test instances of test set *TC*1.

	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/minmax/vi-B&C	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
+On(55)	17.2	7.3	3.3	19.1	6.5	2.2	9.4	9.2	5.1	13.4
+On(70)	101.9	39.6	12.0	117.1	30.6	5.8	53.2	49.2	23.8	78.8
+On(85)	268.0	144.3	58.0	324.2	118.6	27.5	158.9	154.6	102.7	210.8
+Off1(55)	1.4	0.6	0.4	0.5	1.0	0.8	0.6	0.9	0.7	0.9
+Off1(70)	1.8	0.9	0.5	0.6	1.2	1.4	0.9	1.3	0.9	1.2
+Off1(85)	2.3	1.3	0.6	0.6	1.4	2.1	1.2	1.6	1.3	1.5

Table 7-50: Average cost increase [%] (upper bound: best solution found within 3 minutes; lower bound: optimal solution to PLSP/all/vi-B&C without any restrictions on utilization) for test instances of test set *TC*1.

	Capacity utilization			TBO profile			Demand variation		Setup time length	
	low	med.	high	low	med.	high	low	high	low	high
PLSP/all/vi-B&C	0.1	0.2	0.3	0.0	0.1	0.4	0.2	0.1	0.1	0.2
+On(55)	10.7	4.8	2.3	11.0	4.4	2.3	5.6	6.2	3.5	8.3
+On(70)	66.4	24.7	7.4	73.4	19.5	5.5	33.7	31.9	14.4	51.2
+On(85)	192.1	98.4	35.8	219.7	84.4	22.2	110.0	107.5	69.6	148.0
+Off1(55)	1.3	0.8	0.5	0.2	0.8	1.5	0.8	0.9	0.7	1.0
+Off1(70)	1.8	1.0	0.6	0.3	0.9	2.2	1.1	1.2	0.9	1.4
+Off1(85)	2.2	1.3	0.7	0.4	1.0	2.9	1.4	1.5	1.2	1.6

From these results, one can conclude that restrictions requiring a minimal resource utilization have only a minor impact on solutions as long as resources are allowed to be switched off. If this is not an option in reality, because production has to continue all time at a certain rate, the model needs to be amended to guide the selection (and quantity) of products that are produced in excess of demand.

7.4.3 Computational Performance

The computational cost of having additional restrictions on required minimal resource utilization levels is analyzed in this section. All computational results have been obtained using XpressMP release 2003G with standard settings as a MIP solver on a PC equipped with a Pentium IV processor with a clockspeed of 1.7 GHz and 256 MB of memory.

Table 7-51 shows average and maximum gaps after 3 minutes of computational time for the different required minimum resource utilization levels and the different variants to model minimum resource utilization. Although slightly higher than the gaps of model formulations without any restrictions on minimal resource utilization levels, the difference is rather small.

For resources with off times, the average gap increases, if the required minimal resource utilization increases. No such pattern is identified for resources which are not allowed to be switched off ("On"). There, the scenario with a minimal resource utilization level of 70% shows the highest average gaps.

Table 7-51: Average and maximum gap [%] (upper bound: best solution found within 3 minutes; lower bound: bound after 3 minutes of computation) for test set *TC*1.

Average Gap	On			Off1			Off2			None
Resource utilization ≥	55%	70%	85%	55%	70%	85%	55%	70%	85%	0
PLSP/–/vi-B&C	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.1	0.3	0.0
PLSP/minmax/vi-B&C	0.5	0.7	0.6	0.6	0.7	0.8	0.8	1.0	1.1	0.3
PLSP/all/vi-B&C	2.7	3.3	2.8	2.7	2.9	3.1	2.9	3.1	3.5	1.7

Maximum Gap	On			Off1			Off2			None
Resource utilization ≥	55%	70%	85%	55%	70%	85%	55%	70%	85%	0
PLSP/–/vi-B&C	4.1	5.0	2.6	7.2	3.7	4.5	3.1	2.9	5.4	2.6
PLSP/minmax/vi-B&C	11.4	10.9	6.9	9.4	11.9	9.9	11.6	10.1	10.2	8.4
PLSP/all/vi-B&C	15.1	18.5	15.0	15.7	14.0	19.2	17.8	15.0	16.5	13.5

Fig. 7-18 shows the number of test instances solved to proven optimality over time. If no restrictions on required minimal resource utilization levels are present, almost all test instances (170 out of 180) are solved to optimality within the given time limit. If minimum resource utilization levels have to be obeyed, fewer, but still a majority of test instances are solved to proven optimality (142, 147 and 152). The three variants show a similar behavior, which does not change if other minimal resource utilization levels are chosen.

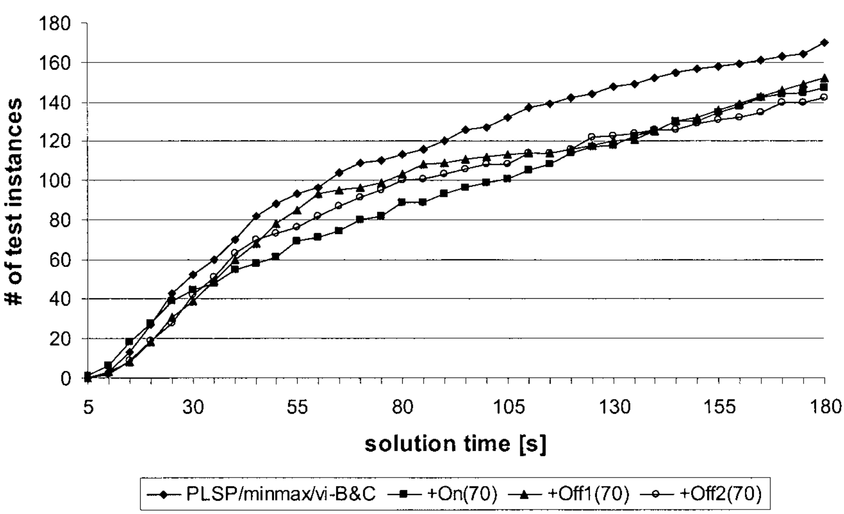


Fig. 7-18: Number of test instances solved to optimality over time (different model formulations, test set *TC*1).

Fig. 7-19 shows what will happen, if batch size restrictions need to be obeyed in addition to required minimal resource utilization levels. Again, all model formulations show a similar behavior, but there is one exception. The exception concerns the case that resources are not allowed to be switched off. In this scenario, fewer test instances can be solved to proven optimality, if the required minimal resource utilization level is higher.

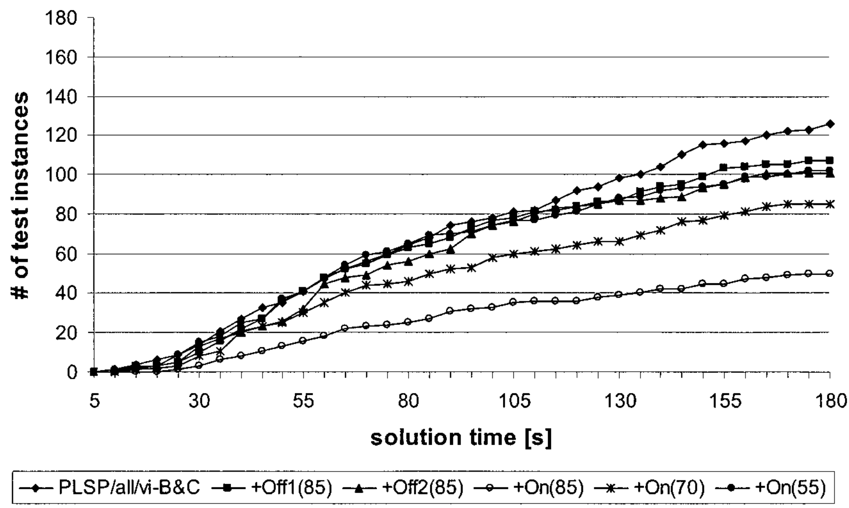


Fig. 7-19: Number of test instances solved to optimality over time (different model formulations, test set *TC*1).

Summing up, to include requirements of minimal resource utilization seems to have low computational costs. Average gaps after 3 minutes of computational time are only slightly higher compared to model formulations without any such restrictions based on test set *TC*1. Thus, the solution quality stays almost the same (Table 7-51). Only if resources are not allowed to be switched off, it seems more difficult to prove optimality (Fig. 7-19).

7.5 Further Extensions

This section explores how further extensions can be incorporated into the proposed modeling and solution approach. Therefore, a standard test set "*KM*" is chosen, which has been used by several authors before,[359] and the model formulation is adapted accordingly. The extensions considered in test set *KM* comprise

- penalties on backlog,
- penalties on safety stock target violations,
- sequence dependent setup costs,

[359] Karimi and McDonald (1997), Ierapetritou et al. (1999) and Lee et al. (2002).

- variable production rates,
- parallel resources and
- planning of maintenance and test tasks within a given time interval.

7.5.1 Test Set and Benchmarks

The test set *KM* has been introduced first by Karimi and McDonald (1997)[360] and describes a chemical site consisting of seven resources. Fourteen different products are produced on the site, but not all products can be produced on each resource. The aim is to derive a production schedule for the next three months, which minimizes a cost function which includes inventory holding costs, sequence dependent setup costs as well as penalties on backlogs and violations of product dependent safety stock targets. The production schedule needs to be more accurate for the first month and is therefore derived in weekly buckets (7 days) for the first month and in monthly buckets (30 days) thereafter.

In Table 7-52 the initial conditions of the plant are given. For each resource the initial setup state and the duration of the campaign prior to the beginning of the planning interval are provided. For each product on each resource the minimal campaign length is 10 days. As only resource m=4 is allocated for a shorter time all other resources are free to change their setup state at the beginning of the planning interval.

Apart from production, on some resources several other tasks (test of equipment and maintenance operations), which will be called outages, have to be carried out. These outages are defined by a fixed duration and may only be carried out in certain time windows (see also Table 7-52).

Table 7-52: Initial conditions and planned outages for test set *KM* (Karimi and McDonald (1997), p. 2709, table 1).

resource	initial conditions		planned outages		
m	product j	duration [days]	outage	duration [hours]	time window [days]
1	3	20			
2	none	10			
3	1	35	test1	100	[15; 30]
			test2	200	[30; 60]
4	7	8			
5	12	15	maintenance	400	[15; 88]
6	12	16			
7	8	30	test3	300	[60; 88]

Table 7-53 provides the initial inventory levels at the beginning of the planning interval as well as safety stock targets and product demands. Product demands have to be fulfilled at the end of each period. Safety stock targets are watched monthly. Therefore, penalties for the violation of safety stock targets are only cal-

360 Cf. Karimi and McDonald (1997) pp. 2709-2712.

culated at the end of week 4, month 2 and month 3. The penalty cost data is given in Table 7-54. Penalties on backlog on the other hand are assessed in each period.

Table 7-53: Initial inventories, safety stock targets and demands for test set *KM* (Karimi and McDonald (1997), p. 2710, table 2).

product	stocks [mass]		product demands [mass]					
j	initial	safety	week 1	week 2	week 3	week 4	month 2	month 3
1	10700	2000	1400	15000	1400	15000	5600	60000
2	1060	1060	775	775	775	775	3100	3100
3	1080	1080	10350	8150	10350	8150	41400	32600
4	1080	1080	2075	1825	2075	1825	8300	7300
5	20030	2000	9725	10025	9725	10025	38900	40100
6	1080	1080	4600	5800	4600	5800	18400	23200
7	1080	1080	10150	9575	10150	9575	40600	38300
8	1080	1080	3550	4000	3550	4000	14200	16000
9	1080	1080	325	425	325	425	1300	1700
10	1080	1080	4125	3725	4125	3725	16500	14900
11	1080	1080	2475	4500	2475	4500	9900	18000
12	2060	2060	1575	1325	1575	1325	6300	5300
13	1080	1080	3550	4000	3550	4000	14200	16000
14	1080	1080	1250	1250	1250	1250	5000	5000

Furthermore, Table 7-54 provides data on holding costs. In this test set it is assumed that demand is fulfilled at the end of each period, but at the same time inward stock movement is continuously. Therefore, holding costs are calculated by assessing inventory at the beginning of a period with a full holding cost coefficient and the production quantity of a period with half of the holding cost coefficient.

Table 7-54: Shortage penalties, inventory holding cost and production rates for test set *KM* (Karimi and McDonald (1997), p. 2710, table 3 and 4).

product	shortage penalties [$/mass]		holding cost	resources and production rates [mass/hour]					
j	safety stock	backlog	[$/1000 mass/day]	m	$minrate_{mj}$	$maxrate_{mj}$	m	$minrate_{mj}$	$maxrate_{mj}$
1	1.7	5.1	1.304	3	11.0	21.9			
2	1.6	4.8	1.227	3	14.3	28.6	4	71.4	114.2
3	1.8	5.4	1.381	1	61.6	123.3			
4	1.8	5.4	1.381	2	50.8	101.5			
5	2.3	6.9	1.764	5	4.8	12.0	6	4.8	12.0
6	1.8	5.4	1.381	1	75.2	150.3			
7	1.8	5.4	1.381	3	12.4	24.7	4	61.9	99.0
8	1.8	5.4	1.381	7	31.4	52.3			
9	1.8	5.4	1.381	7	25.8	43.0			
10	1.8	5.4	1.381	2	51.3	102.6			
11	1.8	5.4	1.381	1	66.3	132.5			
12	2.7	8.1	1.995	5	4.8	12.0	6	4.8	12.0
13	1.8	5.4	1.381	7	29.1	48.5			
14	1.8	5.4	1.381	3	11.7	23.4	4	58.6	93.8

In Table 7-54 the assignment of products to resources as well as their respective minimal and maximal production rates are given. Each product can be produced on two resources at most. By analyzing this information, it is possible to decompose the overall production planning problem into five subproblems defined by resource groups, such that each product belongs to only one single resource group. Thereby, the resource groups are independent from each other and the associated planning problems may be solved separately. Resource m=1 (m=2; m=3 and m=4; m=5 and m=6; m=7) makes up the first resource group and will be called test instance *KM*1 (*KM*2; *KM*3; *KM*4; *KM*5) in the rest of this section 7.5.

Four different kinds of setup operations have to be distinguished. There is no cost incurred to shut down a resource for doing nothing (idle time), test or maintenance, whereas a setup cost of 5,000$ is assessed for starting regular production after idling, test or maintenance. Some setup operations are fairly easy and cost only 600$. These are the setup operations from product j=6 to j=3 (3→11, 4→10, 8→13, 12→5 and 7→14).[361] Setup costs between any two other regular products are 3,000$ each. Setup times are neglected, because they are considerably shorter than the minimal campaign length of 10 days.

Three model formulations from literature will be used to benchmark the proposed modeling and solution approach. These are the model formulations by Karimi and McDonald (1997, model M2), Ierapetritou et al. (1999) and Lee et al. (2002).[362]

7.5.2 Customization of Solution Approach

As test set *KM* assumes sequence dependent setup costs, a basic model formulation has to be chosen which is capable of determining not only lot sizes but also the production sequence. Therefore the PLSP is chosen here as a basis.[363] Furthermore, minimal lot sizes (campaign quantities) have to be observed. As production runs can span over several periods, a model formulation that considers time continuity with respect to lot sizes needs to be chosen.[364] Here, it suffices to select the most simple variant ((6-18)–(6-22) and (6-28)), because there are no restrictions on the maximal lot size.

As there are several specialties to be considered the complete customized model formulation will be presented here. Idle time (j=0) and outages are modeled as additional products. Outages are modeled as products, for which backlog is not allowed, production is only possible during the assigned time windows (Table 7-52) and the minimal (and maximal) campaign length equals the duration.

Moreover, a different time discretization needs to be chosen compared to the problem statement, because the basic model PLSP allows at most one setup opera-

361 Karimi and McDonald (1997, p. 2710) add also 7→8 and 8→14, but these products are not produced on the same resource.

362 The model formulations can be found in the appendix.

363 See sections 2.3.3 and 6.1.1.1.

364 See section 6.1.2.1.

tion per period. As the minimal campaign length is 10 days, it generally suffices to split the monthly time buckets (month 2 and month 3) in three buckets of 10 days size. The reasoning for this is, that within each 10-day bucket at most one setup operation can occur. As one of the test tasks (test 1 on m=3) lasts only for 4.17 days and to make the time discretization generally applicable, the following algorithm is applied to derive the time structure:

> Each time period $s \in \mathcal{T}^M$ (with $\mathcal{T}^M$ denoting the given time structure) is divided into r time periods of equal length with r being the smallest integer value bigger than the capacity (=duration) of time period s divided by the minimal campaign length of any product producible in period s.[365]

In the model formulation the index s is used to refer to the time structure of the initial problem statement, whereas the index t is used to refer to the time structure derived by the algorithm described above.

$$Min \quad \sum_{j\in\mathcal{J}}\sum_{s\in\mathcal{T}^M} h_{js}\cdot I_{js-1} + \sum_{j\in\mathcal{J}}\sum_{s\in\mathcal{T}^M}\sum_{t\in\mathcal{T}_s}\sum_{\substack{m\in\mathcal{M}\\ \wedge j\in\mathcal{J}_m}} {h_{js}}/{2}\cdot X_{mjt}$$
$$+\sum_{i\in\mathcal{J}}\sum_{j\in\mathcal{J}}\sum_{\substack{m\in\mathcal{M}\\ \wedge j\in\mathcal{J}_m\\ \wedge i\in\mathcal{J}_m}}\sum_{t\in\mathcal{T}} sc^{sd}_{mij}\cdot Y_{mijt} + \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} ssp_{jt}\cdot ISSV_{jt} + \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} blp_{jt}\cdot IB_{jt} \qquad (7\text{-}1)$$

$$IB_{jt} + I_{jt-1} + \sum_{\substack{m\in\mathcal{M}\\ \wedge j\in\mathcal{J}_m}} X_{mjt} = I_{jt} + IB_{jt-1} + d_{jt} \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T} \qquad (7\text{-}2)$$

$$ISSV_{jt} \geq sst_j - I_{jt} \qquad \forall j\in\mathcal{J},\ t\in\mathcal{T} \qquad (7\text{-}3)$$

$$\sum_{j\in\mathcal{J}_m} a_{mj}\cdot XT_{mjt} = c_{mt} \qquad \forall\ m\in\mathcal{M},\ t\in\mathcal{T} \qquad (7\text{-}4)$$

$$minrate_{mj}\cdot XT_{mjt} \leq X_{mjt} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m,\ t\in\mathcal{T} \qquad (7\text{-}5)$$

$$X_{mjt} \leq maxrate_{mj}\cdot XT_{mjt} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m,\ t\in\mathcal{T} \qquad (7\text{-}6)$$

$$XT_{mjt} \leq \frac{c_{mt}}{a_{mj}}\cdot (Z_{mjt} + Z_{mjt-1}) \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m,\ t\in\mathcal{T} \qquad (7\text{-}7)$$

$$Y_{mijt} \geq Z_{mjt} + Z_{mit-1} - 1 \qquad \forall\ m\in\mathcal{M},\ i,j\in\mathcal{J}_m, i\neq j,\ t\in\mathcal{T} \qquad (7\text{-}8)$$

$$\sum_{j\in\mathcal{J}_m} Z_{mjt} = 1 \qquad \forall\ m\in\mathcal{M},\ t\in\mathcal{T} \qquad (7\text{-}9)$$

$$K_{mjt} \leq K_{mjt-1} + XT_{mjt} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m\backslash\{0\},\ t\in\mathcal{T} \qquad (7\text{-}10)$$

---

365 E.g., in example *KM3* week 3 (in days:]14; 21]) has a capacity (duration) of 7 days. The minimal campaign quantity of any product producible on any resource in *KM3* is 4.17 (test 1 on m=3). Therefore week 3 is split into two periods ($r = \lceil 7/4.17 \rceil = \lceil 1.68 \rceil = 2$) of equal length (3.5 days).

$$K_{mjt} \geq K_{mjt-1} + XT_{mjt} - \sum_{\substack{i \in \mathcal{J}_m \\ i \neq j}} maxlot_{mj} \cdot Y_{mijt} \qquad \forall m \in \mathcal{M}, j \in \mathcal{J}_m \setminus \{0\}, t \in \mathcal{T} \setminus \{T\} \tag{7-11}$$

$$K_{mjt} \leq maxlot_{mj} \cdot (1 - \sum_{\substack{i \in \mathcal{J}_m \\ i \neq j}} Y_{mijt+1}) \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m \setminus \{0\}, t \in \mathcal{T} \setminus \{T\} \tag{7-12}$$

$$K_{mjt-1} + XT_{mjt} \geq minlot_{mj} \cdot \sum_{\substack{i \in \mathcal{J}_m \\ i \neq j}} \sum_{\substack{k \in \mathcal{J}_m \\ k \neq i}} Y_{mikt} \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m \setminus \{0\}, t \in \mathcal{T} \tag{7-13}$$

$$\sum_{\substack{i \in \mathcal{J}_m \\ i \neq j}} Y_{mijt} \leq 1 - Z_{mjt-1} \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m, t \in \mathcal{T} \tag{7-14}$$

$$\begin{aligned} &I_{jt} \geq 0, \quad IB_{jt} \geq 0, \quad IB_{j0} = 0, \\ &K_{mjt} \geq 0, \quad X_{mjt} \geq 0, \quad XT_{mjt} \geq 0, \\ &Y_{mijt} \geq 0 \end{aligned} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \tag{7-15}$$

$$Z_{mjt} \in \{0;1\} \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m, t \in \mathcal{T} \tag{7-16}$$

Indices and index sets:

s	Periods, $s \in \mathcal{T}^M$
$\mathcal{T}^M$	Set of (macro-)periods defined by due dates
$\mathcal{T}_s$	Set of periods which belong to (macro-)period s

Data:

blp_{jt}	Backlog penalty cost associated with one unit of backlog of product j at the end of period t
$maxlot_{mj}$	Maximal lot size (campaign quantity) for product j on resource m
$maxrate_{mj}$	Maximal production rate of product j on resource m
$minlot_{mj}$	Minimal lot size (campaign quantity) for product j on resource m
$minrate_{mj}$	Minimal production rate of product j on resource m
ssp_{jt}	Safety stock penalty cost associated with the violation of the safety stock target for one unit of product j at the end of period t
sst_j	Safety stock target for product j

Variables:

IB_{jt}	Backlog of item j at the end of period t
$ISSV_{jt}$	Safety stock violation for item j at the end of period t
K_{mjt}	Campaign variable for product j on resource m in period t (current campaign quantity up to period t)
X_{mjt}	Production quantity of product j on resource m in period t
XT_{mjt}	Production time of product j on resource m in period t
Y_{mijt}	Setup variable (=1, if a setup operation from item i to item j is performed on resource m in period t, =0 otherwise)
Z_{mjt}	Setup state variable (=1, if item j is set up on resource m at the end of period t, =0 otherwise)

The objective (7-1) is the minimization of costs. Relevant cost components are inventory holding costs (term 1 and 2), sequence dependent setup costs (term 3) as well as penalties on safety stock target violations (term 4) and backlog (term 5). Inventory balance constraints (7-2) are altered to take the possibility of backlog into account and (7-3) are used to calculate violations of the safety stock target.

As idle time is modeled as a distinct product here, total capacity in each period must be assigned completely to the products producible on the respective resource. Otherwise it would not be possible to account for the setup costs correctly, because these differ if two regular products are produced one after another depending on whether there is idle time between them or not. This is done by modeling capacity constraints (7-4) as equations.

Constraints (7-5) and (7-6) form relationships of production variables indicating the quantity produced (X_{mjt}) and the time produced (XT_{mjt}). Both types of variables are necessary, because the first one is needed to fulfill demands (with mass used as unit of measurement), whereas the second one is used to comply with production requirements (minimal campaign length with time being the unit of measurement). In contrast to the basic model assumptions[366], here the intensity of production is variable between certain bounds (minimal and maximal production rates, Table 7-54).

Constraints (7-7)–(7-9) are taken from the basic PLSP model formulation.[367] They limit production in a certain period to those two products that are set up either at the beginning of the period or at the end of the period (7-7). A setup operation from product i to product j has occurred on resource m in period t (Y_{mijt}=1) if i was set up at the beginning of the period (Z_{mit-1}=1) and j at the end of the period (Z_{mjt}=1) (7-8). Of course, at each period boundary only one product can be setup (7-9).

Time continuity regarding lot sizes is introduced by constraints (7-10)–(7-14), which are essentially the same as constraints (6-18)–(6-22) and (6-28) in section 6.1.2.1. The first two constraints accumulate production (time) for product j on resource m as long as no new setup operation for j on m is performed, while (7-12) resets this variable if this is the case. Whenever any setup operation occurs on resource m the minimal campaign length has to be met (7-13). Again, constraints (7-14) are to prevent unnecessary setup operations.

Finally, (7-15) and (7-16) state the domain of the variables. Note, that variables Y_{mijt} do not need to be defined as binary explicitly. Fractional values of Y_{mijt} will not occur. They are forbidden for setup operations to product j in period t, if j is set up at the beginning of t (7-14). On the other hand, constraints (7-8) force Y_{mijt} to "1" in case a setup operation takes place.

In addition, some variables need to be initialized at the beginning of the planning interval. This is done for variables Z_{mjt} and K_{mjt} according to the initial plant state given in Table 7-52. Furthermore, some variables can be fixed or bounded based on the problem data. These are the variables that model outages. As these are only allowed in certain time windows, binary setup state variables Z_{mjt} outside

366 See sections 2.2–2.4 and 6.1.1–6.1.5.

367 See sections 2.3.3 and 6.1.1.1.

these time windows may be fixed to zero. Furthermore, if an outage can lie within a period only to a certain extent, this is modeled by providing an upper bound on the corresponding production time variable XT_{mjt}.[368]

Similar to valid inequalities (6-3) and (6-4) in section 6.1.1.1, two types of valid inequalities might be added to the model formulation. Valid inequalities (7-18) are resource dependent variants of (6-4), whereas (7-17) extend the concept of (6-3) to a model formulation in which backlogging is allowed.

$$I_{jt-1} + IB_{jt+p} \geq \left(\sum_{s=t}^{t+p} d_{js} \right) \cdot (1 - \sum_{\substack{m \in \mathcal{M} \\ \wedge j \in \mathcal{J}_m}} Z_{mjt-1} - \sum_{\substack{m \in \mathcal{M} \\ \wedge j \in \mathcal{J}_m}} \sum_{\substack{i \in \mathcal{J}_m \\ i \neq j}} \sum_{s=t}^{t+p} Y_{mijs}) \quad \begin{array}{c} \forall j \in \mathcal{J}, \\ t=1..T-1, \\ p=1..T-t \end{array} \quad (7\text{-}17)$$

$$Y_{mijt} \leq Z_{mjt} \quad \forall\, m \in \mathcal{M},\, i,j \in \mathcal{J}_m, i \neq j,\, t \in \mathcal{T} \quad (7\text{-}18)$$

Valid inequalities (7-17) state, that if a product j cannot be produced within the time interval [t; t+p] (second parenthesis on the right-hand side evaluates to "1"), the total demand within this interval (first parenthesis on the right-hand side) must either reside in stock at the beginning of the interval ($I_{jt-1} \geq 0$) or is backlogged until the end of the interval ($IB_{jt+p} \geq 0$) or a mixture of both. However, in the rather small test instances considered here, the definition of these two types of valid inequalities did not make a difference and therefore they have not been used in the computational tests.

As will be seen from the analysis of solutions and the evaluation of the computational performance, the proposed customized model formulation will produce optimal solutions as long as the following two assumptions are met. First, the minimal campaign length needs to be greater than the period length, which is the case here due to the time discretization procedure outlined above. Second, as idle time is modeled as the production of a dummy product, idle time is counted as a product with respect to the basic PLSP assumption, which states that at most one setup operation might be performed in each period.[369] Although idle time will not obey any minimal lot size restrictions ((7-10)–(7-13)), this prevents short amounts of idle time between two production runs of regular products. Anyhow, as production intensity may be shifted within rather wide ranges (Table 7-54) and because setup operations starting from the idle state are rather expensive, this will not lead to any problems in finding the optimal solutions here.

7.5.3 Analysis of Solutions

Table 7-55 provides the optimal objective function values (and its components) for the five test instances of test set *KM*, while Fig. 7-20 provides gantt charts of the optimal solutions. These gantt charts are not distinct, because production rates

---

[368] E.g., in example *KM3* at most two days of outage test 2 (time window: [30; 60]) can lie in month 3 (in days:]58; 88]). Therefore an upper bound for the corresponding variable XT_{mjt} is 2. Cf. Lee et al. (2002) pp. 62-63.

[369] See also section 2.3.3.

are allowed to vary in rather wide ranges and a different allocation of time to products will yield the same objective function value as long as the same quantities are produced in each period.

Table 7-55: Optimal objective function values for test set *KM*.

example	optimal objective function value [$]	holding costs [$]	setup costs [$]	safety stock violation penalty [$]	backlog penalty [$]
*KM*1	128,003	4,889	8,400	0	114,713
*KM*2	16,704	2,731	8,600	0	5,373
*KM*3	350,816	4,713	16,600	10,200	319,303
*KM*4	797,129	1,495	4,800	30,486	760,348
*KM*5	43,272	2,301	12,200	9,720	19,052

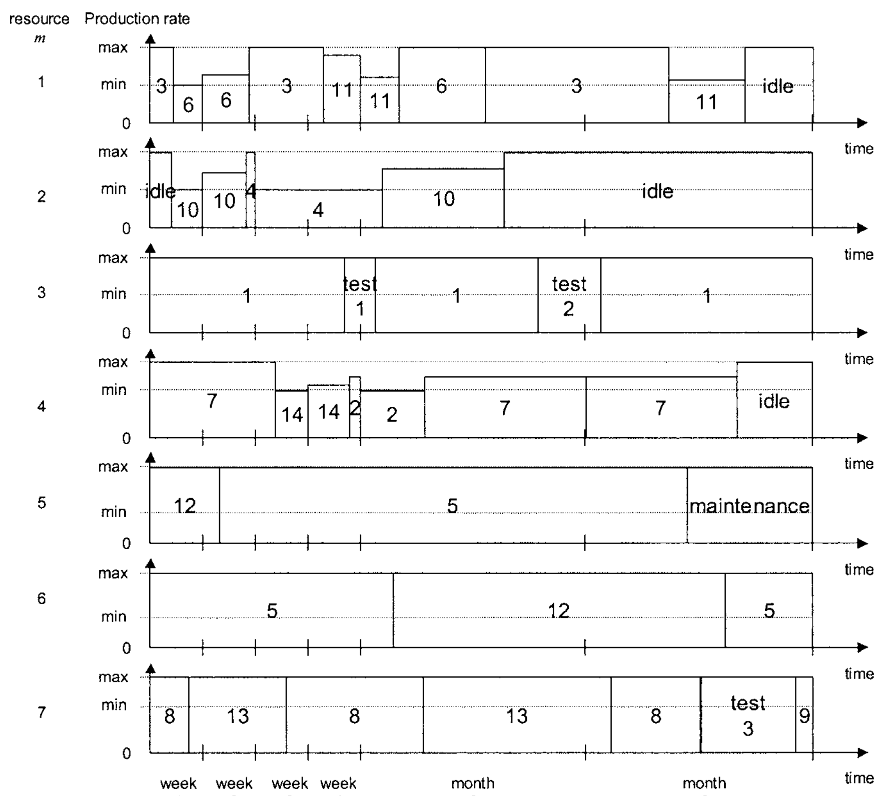


Fig. 7-20: Gantt charts of optimal solutions for test set *KM*.

However, for some test instances, different optimal solutions have been reported in literature. These differences will be analyzed first, before the proposed model formulation is compared computationally to the benchmarks by Karimi and McDonald (1997), Ierapetritou et al. (1999) and Lee et al. (2002).

For test instance *KM*1, Karimi and McDonald (1997) and Ierapetritou et al. (1999) report a different objective function value of 125,603$.[370] However, they report the same optimal solution (gantt chart).[371] Thus, they might have used a slightly different data set. Their model formulations have been rerun with our data set yielding an optimal objective function value as reported in Table 7-55.

With respect to test instance *KM*2, all three benchmarks propose different optimal objective function values. Karimi and McDonald (1997) report 16,201$, Ierapetritou et al. (1999) report 16,138$, while Lee et al. (2002) report 16,737$.[372] However, the last two propose the same optimal solution (gantt chart), while in the optimal solution of Karimi and McDonald (1997) the setup operation from product 4 to product 10 is scheduled to be in the fourth week instead of the second month. Anyhow, neither of the reported optimal solutions is the correct solution with respect to the problem data, because all solutions start with production of product 10 at the beginning of the first week. However, as the resource is idle at the beginning of the planning interval, it is possible to save holding costs by keeping the resource idle for some more days.

Karimi and McDonald (1997) could not find the optimal solution with their model formulation, because one of their constraints forbids idle time, if backlogging occurs in a later period.[373] Thus, in this test instance, idle time in the first week is not allowed by their model formulation. However, in some rare cases (like this one) it might be less costly to keep a resource idle, because the product that is backlogged is not produced in their solution either. If this constraint is removed from the model formulation, this leads to the same optimal objective function value and solution (gantt chart) as reported here. We could not reproduce the objective function values of the other two papers, because in our implementation, these model formulations produced the optimal objective function value and optimal solution as reported here.

For test instance *KM*3, again all three benchmarks propose different optimal objective function values. Karimi and McDonald (1997) report 350,257$, Ierapetritou et al. (1999) report 350,216$, while Lee et al. (2002) report 345,944$.[374] Ierapetritou et al. (1999) report the same optimal solution as provided in Fig. 7-20. Thus, their reported optimal objective function value, which differs from ours by 600 $ might be due to a different setup cost coefficient used. Moreover, with our implementation of their model formulation we arrived at 350,816$ as reported in Table 7-55. The model formulation by Karimi and McDonald (1997) seems to suf-

---

370 Cf. Karimi and McDonald (1997) p. 2711 and Ierapetritou et al. (1999) p. 3459.

371 Here (and for test instances *KM*2–*KM*5), the optimal solutions are considered the same, if production quantities are the same in each period. Due to the variation of production rates, allocation of time to products may vary without altering the optimal objective function value.

372 Cf. Karimi and McDonald (1997) p. 2711, Ierapetritou et al. (1999) p. 3459 and Lee et al. (2002) p. 62.

373 Cf. Karimi and McDonald (1997) pp. 2705, 2706, 2714.

374 Cf. Karimi and McDonald (1997) p. 2711, Ierapetritou et al. (1999) p. 3459 and Lee et al. (2002) p. 62.

fer from the same constraint as in test instance *KM2*, although we could not reproduce their solution. They propose to start to produce product 14 again at the end of the third month, presumably because no idle time is allowed at the end of the third month, because a backlog (for product 1) occurs. Moreover, the difference of 41$ between Karimi and McDonald (1997) and Ierapetritou et al. (1999) is exactly the difference in holding costs of product 14 in their respective solutions. However, the additional setup cost cannot be accounted for by this explanation.

The objective function value and solution reported by Lee et al. (2002) are definitely wrong. First, they propose to schedule test 2 directly after test 1. Postponing it like in the solution proposed in Fig. 7-20 would save 5,360$ in backlog costs, but incur only 5,000$ in additional setup costs. Moreover, they produce too much of product 7 in the third month with an ending inventory of 7,348 exceeding the safety stock target of 1,080. Producing less, which is possible as the minimal campaign length has already been met, would reduce holding costs by 130$. Our implementation of their model formulation produces the correct optimal objective function value and solution, after some initialization conditions have been added.

With respect to test instance *KM4*, again all three benchmarks propose different optimal objective function values. Karimi and McDonald (1997) report 794,385$, Ierapetritou et al. (1999) report an optimal objective function value of 794,386$, while Lee et al. (2002) report 800,278$.[375] The optimal solution (gantt chart) by Karimi and McDonald (1997) matches ours except for the first lot of product 12 being produced on resource 6 instead of resource 5. As both resources are identical (Table 7-54), this does not change the optimal objective function value. Thus, they might have used a somewhat different data set, because in our implementation their model formulation yields the same optimal objective function value as reported above (Table 7-55).

The optimal solution (gantt chart) by Ierapetritou el al. (1999) shows production of product 12 on both resources at the beginning of the first week. This is not optimal, because inventory holding costs for product 12 are higher than for product 5. As both products are produced at the same production rate and as no backlogging occurs in the first period, it would have been better to start production of product 5 on one resource right at the beginning of the first week, while postponing some of the production of product 12 into the second week. The cost saving for doing so is exactly 1.03$ and explains the different objective function values reported.

Again, the optimal solution (gantt chart) by Lee et al. (2002) is definitely wrong and may be improved by inspection. In addition to producing product 12 on both resources at the beginning of the first week (like Ierapetritou et al. (1999)), they propose to schedule maintenance at the beginning of the third month instead of the end. This incurs additional setup costs of 5,000$ without yielding any benefits.

---

[375] Cf. Karimi and McDonald (1997) p. 2711, Ierapetritou et al. (1999) p. 3459 and Lee et al. (2002) p. 62.

The reason for this is that constraints (23) of their paper does not allow outages at the end of the planning horizon.[376]

For test instance *KM*5, Karimi and McDonald (1997) and Ierapetritou et al. (1999) report a different objective function value of 42,072$.[377] However, they report the same optimal solution (gantt chart). Thus, they might have used a slightly different data set also here. Their model formulations have been rerun with our data set yielding an optimal objective function value as reported in Table 7-55.

Lee et al. (2002) report the same optimal objective function value, but a slightly different optimal solution (gantt chart).[378] In their optimal solution test 3 and production of product 8 are switched in the third month. However, this does not change the objective function value.

Although not affecting optimal solutions to test set *KM*, there is another flaw in the model formulation by Lee et al. (2002). They forgot to model sub-tour elimination constraints.[379] Thus, setup cycles distorting solutions are possible. This is illustrated best by an example: If the setup operation from product 13 to product 9 is easy instead of difficult and therefore costs only 600$ instead of 3,000$, the model formulation by Lee et al. (2002) will produce a setup cycle in the third month. The reason is, that it is cheaper to perform setup operations from product 13 to 9 (600$), 8 to test 3 (0$), test 3 to idle (0$) and idle to 8 (5,000$) totaling 5,600$ than performing the optimal sequence from 13 to 8 (3,000$), 8 to test 3 (0$), test 3 to 9 (5,000$) costing 8,000$. The first setup pattern is not forbidden, because sub-tour elimination constraints are missing in their model formulation. However, the sequence cannot be implemented. Therefore, the optimal solution of the model formulation by Lee et al. (2002) is not even feasible, because of the missing link in and out of the cycle (8, test 3, idle).

---

376 Moreover, if they had modeled idle time in this test instance, they would have received the correct schedule, because at the end of the planning interval a setup operation from maintenance to idle would have occurred at a cost of 0$, which in fact would have allowed the maintenance at the end of the planning horizon despite the error in constraint (23). However, they modeled idle time only for those test instances from which they knew that idle time would occur in the optimal solution. This is obvious from the solution statistics presented in their paper and understates the true number of binary variables necessary for modeling, because the optimal solution is usually not known beforehand. E.g., they claim that only 54 binary variables have been used to model test instance *KM*4. On each resource there are 3 products to be produced (5, 12 and idle). With 6 periods and 2 variables for each product on each resource, this gives $3 \cdot 2 \cdot 6 \cdot 2 = 72$ binary variables. The binary variables for the maintenance operation have to be defined only for the periods and resource the maintenance operation is allowed to be scheduled ($1 \cdot 1 \cdot 4 \cdot 2 = 8$). Thus, in our view, 80 binary variables are necessary in the model formulation by Lee et al. (2002) for test instance *KM*4. Presumably, they did the same with test instances *KM*1 and *KM*5.

377 Cf. Karimi and McDonald (1997) p. 2711 and Ierapetritou et al. (1999) p. 3459.

378 Cf. Lee et al. (2002) pp. 62-64.

379 See constraints (2-42) for the CLSD (section 2.4.2).

7.5.4 Computational Performance

As has been outlined in the previous section 7.5.3, all benchmark model formulations had minor defects. These have been removed, such that all model formulations will find the same optimal solutions and can be compared on a fair basis.[380] XpressMP release 2003G with standard settings has been used as a MIP solver on a PC equipped with a Pentium III processor with a clockspeed of 800 MHz and 128 MB of memory to obtain the computational results in this section.

In Table 7-56 the size of the model formulations (number of binary variables, total number of variables, number of constraints and number of non-zero coefficients), lower bounds provided by the model formulations as well as the time to prove optimality and the size of the branch-and-bound tree necessary to do this (number of nodes) are compared for the five test instances of test set *KM*.

Except for test instance *KM*3 the proposed model formulation (④) needs the fewest number of binary variables to model the described decision situation. If there had not been the missing sub-tour elimination constraints, which require setup operation variables Y_{mijt} to be declared binary, the model formulation by Lee et al. (2002, ③) would have needed the second fewest number of binary variables. On the other hand, their model formulation now produces always the best bounds and therefore the smallest search trees. However, the time needed to prove optimality of solutions is lower for the proposed model formulation as long as the test instance comprises only a single resource (test instances *KM*1, *KM*2 and *KM*5).

The model formulation by Karimi and McDonald (1997, ①) also provides tight lower bounds, but suffers from its bigger model size yielding longer solution times (with test instance *KM*3 being an exception) than the proposed model formulation.

Finally, the model formulation by Ierapetritou et al. (1999, ②) shows the weakest bounds. Consequently (with one exception), the biggest branch-and-bound trees are built by this model formulation. To prove optimality to test instance *KM*4 takes almost half an hour, while all other model formulations only take a few seconds. However, they claim in their paper to have obtained much smaller branch-and-bound trees by assigning priorities for branching on the binary variables.[381] Nothing similar has been done here.

Summarizing the results, the application of the proposed model formulation reveals new insights and a new optimal solution (*KM*2) to a test set frequently used in literature. Computational effort has been slightly higher compared to two of the three benchmarks, if parallel resources need to be planned for. However, no effort has been taken to tailor the proposed model formulation to this scenario. Although only the extension to respect minimal lot sizes could be used here, the proposed model formulation has been shown to be highly competitive. Thus, further additions to the model formulation (here: penalties on backlog and safety stock violations, sequence dependent setup costs, variable production rates, planning of outages) can be made easily without loosing performance.

380 The (corrected) model formulations can be found in the appendix.

381 Cf. Ierapetritou et al. (1999) p. 3458.

Table 7-56: Comparison between the proposed model formulation (④) and the benchmark model formulations by Karimi and McDonald (1997, ①), Ierapetritou et al. (1999, ②) and Lee et al. (2002, ③) for test set *KM*.

Test instance	Model formulation	# binary variables	# variables	# constraints	# non-zero coefficients	# nodes	LB^{XLP} [$]	Optimal solution [$]	Time [s]
*KM*1	①	64	496	556	1,985	147	20,487	128,003	4
	②	45	471	1,067	3,824	130	15,783		3
	③	120	297	489	1,533	85	24,381		2
	④	40	363	460	1,695	250	20,263		1
*KM*2	①	48	321	423	1,353	33	13,452	16,704	1
	②	30	296	596	2,210	48	7,419		0
	③	72	197	325	957	7	16,000		0
	④	30	229	297	939	31	14,938		0
*KM*3	①	158	2,145	1,569	7,190	193	327,987	350,816	24
	②	117	1,084	2,997	10,083	7,187	264,376		120
	③	442	980	1,562	5,219	74	333,025		8
	④	125	1,488	1,918	9,689	1,692	327,016		32
*KM*4	①	107	860	978	3,493	21	795,040	797,129	4
	②	70	528	1226	4,456	383,989	542,647		1,656
	③	188	446	746	2,288	3	797,129		0
	④	67	528	718	2,515	330	791,618		3
*KM*5	①	67	675	703	2,689	453	18,262	43,272	6
	②	51	552	1,294	4,451	1,881	2,676		14
	③	170	371	608	1,974	326	24,278		4
	④	42	517	655	2,743	374	15,740		2

LB^{XLP} is the LP relaxation after automatic cut generation by XpressMP.
A solution time of 0 seconds indicates that the solution was proven optimal in less than one second, because XpressMP rounds solution times down to the second.
The number of slots assigned to individual periods have been two for each weekly period and four for each monthly period in the model formulation of Karimi and McDonald (1997). For the model formulation of Ierapetritou et al. (1999) it has been two for each weekly period and three for each monthly period except for test instance *KM*5, in which four slots have been used in the third month. This is in line with the recommendations of the authors (cf. Karimi and McDonald (1997, p. 2711) and Ierapetritou et al. (1999, p. 3457)) and gives the best results. Generally, several assignments will have to be tested until the "optimal" allocation is known (cf. Karimi and McDonald (1997, p. 2709), Ierapetritou and Floudas (1998a, pp. 4349 and 4358)). Therefore, the "Time" column understates the true computational effort for these model formulations.

7.6 Dependency on Solver Technology

So far, all computational results are based on XpressMP release 2003G as the standard MIP solver. As the main contribution of this thesis is in modeling, this section aims to show that the computational results are rather independent from the MIP solver used. Therefore, a subset of the tests has been run again with three different commercial MIP solvers.

The first one is of course XpressMP release 2003G. The second MIP solver is a newer release of XpressMP (2004B) which became available during the work on this thesis. Third, CPLEX 9.0 has been used. All solvers have been used with standard settings on a PC equipped with a Pentium IV processor with a clockspeed of 2.4 GHz and 256 MB of memory running Windows XP Professional.

Test set *TC*1 and the model formulations using the lot size extensions will be used here again.[382]

Fig. 7-21 shows the number of test instances solved to proven optimality for the model formulation with minimal and maximal lot size restrictions and its corresponding benchmark by Kallrath (1999) for different MIP solvers. For all solvers, the proposed model formulation solves a lot more test instances to proven optimality in the same time. The two different releases of XpressMP show almost the same behavior regardless whether the proposed model formulation or whether the Kallrath (1999) benchmark is used. CPLEX 9.0 clearly outperforms XpressMP, if the benchmark model formulation by Kallrath (1999) is used. With respect to the new model formulation, CPLEX seems to have advantages in the beginning, but needs a lot more time to prove optimality for all test instances. The last test instance is proven optimal after 488 seconds by CPLEX, but after 280 (222) seconds by XpressMP 2004B (2003G).

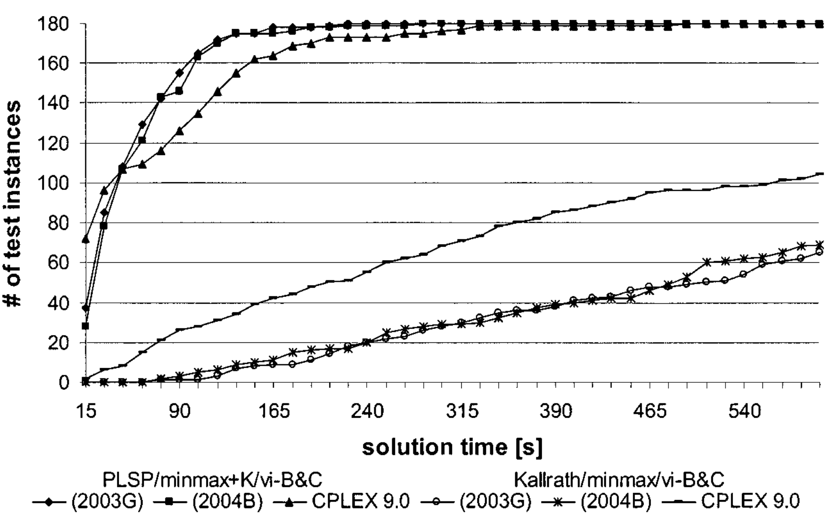


Fig. 7-21: Number of test instances solved to optimality over time (PLSP/minmax+K/vi-C&B and Kallrath/minmax/vi-C&B with different solvers, test set *TC*1).

In Fig. 7-22 the same picture is drawn for model formulations which additionally respect batch size restrictions. Here, not a single test instance can be proven optimal for the benchmark formulation by Kallrath (1999) by either of the three MIP solvers. Concerning the proposed model formulation, all three MIP solvers show a similar behavior. Again, CPLEX seems to be advantageous in the begin-

382 See section 7.2.1 for a description of test instances and Table 7-22 for a description of the notation of the model formulations used here.

ning. More test instances are proven optimal in the first 2 ½ minutes. Afterwards, XpressMP seems stronger. Here the last test instance is proven optimal after 386 (522) seconds by XpressMP 2004B (2003G), while not all test instances (only 177 out of 180) have been proven optimal by CPLEX within the time limit of 10 minutes.

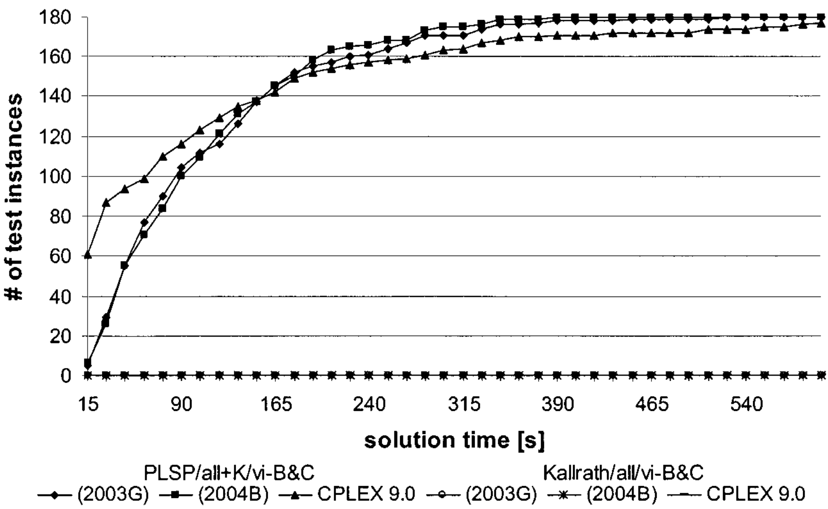


Fig. 7-22: Number of test instances solved to optimality over time (PLSP/all+K/vi-C&B and Kallrath/all/vi-C&B with different solvers, test set *TC*1).

Finally, Table 7-57 compares the solution quality of the different model formulations and solvers. Therefore, the best solutions found within certain time limits are compared to corresponding optimal solutions. Concerning the tests with minimal and maximal restrictions on the lot size, the proposed model formulation provides solutions which deviate from optimal solutions on average only approximately 1% after 15 seconds (MIP solvers: XpressMP 2003G and 2004B). The benchmark model formulation by Kallrath (1999) on the other hand needs more than 60 seconds to find even a feasible solution for all test instances. It takes 10 minutes to obtain a comparable solution quality (~1%) with this model formulation regardless which MIP solver is used.

If batch size restrictions are also needed, the proposed model formulation needs approximately 30 seconds to arrive at the same solution quality as above (~1%). No MIP solver was able to provide solutions to all 180 test instances within the time limit of 10 minutes for the benchmark model formulation by Kallrath (1999). For this model formulation, CPLEX seems to find more and better solutions than XpressMP 2003G which again seems to find more and better solutions than XpressMP 2004B.

For the three other model formulations tested, the ranking of MIP solvers based on the quality of solutions seems to be XpressMP 2004B first, CPLEX 9.0 second and XpressMP 2003G third.

Table 7-57: Solution quality of different model formulations for test sets *TC*1 (upper bound: best solution found within ... seconds; lower bound: optimal solution). Numbers in parenthesis indicate the number of test instances a feasible solution has been found for within the time limit (if not equal to 180).

Solution quality after ... seconds	PLSP/ minmax+K/ vi-C&B	Kallrath/ minmax/ vi-C&B		PLSP/ all+K/ vi-C&B	Kallrath/ all/ vi-C&B	
XpressMP 2003G						
15	1.05%	41.02%	(94)	2.48%	106.83%	(11)
30	0.43%	18.62%	(139)	1.05%	95.04%	(42)
60	0.13%	8.85%	(174)	0.57%	81.43%	(80)
120	0.01%	4.93%		0.18%	62.39%	(105)
180	0.00%	3.47%		0.07%	54.28%	(120)
300	0.00%	2.53%		0.01%	46.42%	(129)
600	0.00%	1.29%		0.00%	40.22%	(141)
XpressMP 2004B						
15	0.95%	17.18%	(137)	1.81%	126.15%	(12)
30	0.39%	11.24%	(165)	0.93%	121.76%	(25)
60	0.17%	5.87%	(176)	0.45%	111.84%	(38)
120	0.03%	3.52%		0.17%	98.88%	(61)
180	0.01%	2.61%		0.07%	97.23%	(78)
300	0.00%	1.90%		0.01%	84.92%	(93)
600	0.00%	0.95%		0.00%	74.59%	(112)
CPLEX 9.0						
600	0.00%	1.06%		0.03%	36.12%	(162)

Summarizing, the difference between the three MIP solvers is much smaller than the difference observed between the proposed model formulations and their benchmarks. Thus, based on these results, the proposed modeling approach seems to be independent from the MIP solver, which has been used as a standard tool throughout this thesis.

8 Summary and Outlook

The aim of this thesis has been to provide a modeling framework to incorporate aspects of time continuity into lot-sizing model formulations based on time discretization.

The starting point has been a classification of basic lot-sizing models. Big-bucket, small-bucket and hybrid models have been compared and their advantages and disadvantages have been discussed. Modeling aspects of time continuity is only possible, if adjacent periods in these models can be linked somehow. Therefore, only small-bucket and hybrid models have been found eligible for further consideration. The PLSP, which is the most versatile small-bucket model, and the CLSPL due to its universality have been chosen as basic models to be used throughout the rest of this thesis.

In the third chapter different extensions to model aspects of time continuity have been defined. The first important cornerstone has been the modeling of setup states at period boundaries. Period boundaries are the result of time discretization, and therefore it is crucial to model these points in time explicitly to avoid disruptions in the resulting plan. With examples from literature it has been shown that plans are fundamentally different depending on whether setup states are carried over period boundaries or not. Later on, this is also shown based on the computational results in section 7.1.

The second cornerstone addressed in chapter three is the modeling of lot sizes that span over two (or more) periods. In the process industries the problem often arises that lot sizes have to respect certain restrictions on their minimal and / or maximal size, or that they have to be in multiples of a predefined batch size. In these scenarios, the correct lot size needs to be determined, and it does not suffice to pose restrictions on the production quantities in each period independently. The modeling of setup operations is the third cornerstone, as setup operations may also lie at the boundary of two periods. Lastly, the impact of these measures, which incorporate time continuity into a bucket-oriented setting, on the resource utilization of the resulting plans has been analyzed.

The fourth chapter has provided a review of relevant literature. The aim of this literature review has been twofold. On the one hand, the emphasis has been on extensions of standard lot-sizing models presented in the second chapter to account for time continuity. On the other hand, model formulations originating from process industries have been discussed. Although these often contain additional side constraints, their thorough analysis has proven worthwhile, as they offer a great variety of modeling ideas, as well as they constitute a major area of application for the model formulations proposed later on. To show the extensibility of the proposed model formulation, it has been benchmarked to three of these model formulations from literature in section 7.5.

Although the major contribution of this thesis lies in modeling, the model should be integrated into a planning framework. Therefore, chapter five introduces briefly the concept of Advanced Planning Systems, for which the proposed model formulations and solution procedure is well-suited. Furthermore, an overview of the state-of-the-art in mathematical programming (including capabilities of current standard solvers) and an introduction into decomposition has been given.

Chapter six contains the major contribution of this thesis. Model formulations for the four cornerstones defined in chapter three have been developed. The model formulations have been presented for both basic models chosen, PLSP and CLSPL. This has been necessary to account for the differences of these two basic models. However, also these two models can be coupled by aligning them at a period boundary.

The extensions to model setup states, lot sizes spanning over several periods, period overlapping setup times and different restrictions on resource utilization have been presented as building blocks and may be freely combined. To enhance computational performance the model formulations are enriched by valid inequalities. As each extension requires several variables and constraints to be added to the model formulation, in order to reduce computational effort usually not a complete model should be solved, but rather one tailored to the individual decision situation. Furthermore, as MIP model formulations often grow too big, a temporal decomposition heuristic has been proposed to cope also with this issue.

Finally, the proposed model formulations and the temporal decomposition heuristic are evaluated computationally in the seventh chapter. An analysis of optimal solutions revealed the fundamental difference of solutions to models which allow for setup state preservation across period boundaries and those that do not (section 7.1.2). Furthermore, it has been shown that minimum or maximum restrictions on lot sizes are considerably easier extensions than batch size restrictions on lot sizes (section 7.2.2). However, the progress in the ability of solving problems containing this kind of extension has been most remarkable. Extensive computational tests have shown that benchmarks from literature are clearly outperformed with respect to solution quality and solution speed (sections 7.1.3 and 7.2.3).

The extensibility of the proposed approach has been demonstrated by solving a test set frequently used in literature and by comparison of its solution to solutions from literature. This analysis not only revealed the computational competitiveness of the proposed approach, but also showed improvements compared to solutions provided in literature (section 7.5). Lastly, the independence of the proposed solution approach from the solver technology used has been shown (section 7.6).

In summary, this thesis has provided a new modeling approach that allows to represent plans that are possible on a continuous time scale within a time-indexed setting. Only small limitations prevail. These are, that in each period at most *one setup operation* (basic model PLSP) or at most *one setup operation per product* (basic model CLSPL) are allowed. On the other hand, the model formulations (and practical relevance) suggest to plan with rather short periods, because otherwise the correct modeling of period boundaries would be less relevant. Thus, by incorporating these measures into a planning system, resulting plans are no longer disrupted by discontinuities due to time discretization. Furthermore, the proposed model formulations have been shown to be computationally efficient.

Further research opportunities in the modeling area could be to calculate holding costs continuously rather than based on end-of-period inventories, and to allow for more options regarding resources to be planned for. These might be parallel resources or other types of (shared) resources like, for example, personnel or tanks. Furthermore, to allow for a multi-level operations structure would be an attractive extension. On the algorithmic side, improving the anticipation capabilities of the temporal decomposition heuristic might be a rewarding topic.

References

Applequist G, Samikoglu O, Pekny J, Reklaitis G (1997): Issues in the use, design and evolution of process scheduling and planning systems. ISA Transactions, 36, 81-121

Ashayeri J, Teelen A, Selen W (1995): Computer integrated manufacturing in the chemical industry: theory & practice. CentER Discussion Paper No. 7, Tilburg University, Tilburg

Atamtürk A, Savelsbergh MWP (2003): Commercial integer programming software systems. Research Report BCOL.03.01, University of California at Berkeley

Barany I, van Roy TJ, Wolsey LA (1984): Strong formulations for multi-item capacitated lot sizing. Management Science, 30, 1255-1261

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

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

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

Bixby RE, Fenelon M, Gu Z, Rothberg E, Wunderling R (2000): MIP: theory and practice – closing the gap. In: Powell MJD, Scholtes S (eds.): System modelling and optimization: methods, theory and applications. Kluwer, Boston, 19-49

Blocher JD, Chand S, Sengupta K (1999): The changeover scheduling problem with time and cost considerations: analytical results and a forward algorithm. Operations Research, 47, 559-569

Blömer F (1999): Produktionsplanung und –steuerung in der chemischen Industrie: Ressourceneinsatzplanung von Batchprozessen auf Mehrzweckanlagen. Deutscher Universitäts-Verlag (Gabler), Wiesbaden

Blömer F, Günther H-O (1998): Scheduling of a multi-product batch process in the chemical industry. Computers in Industry, 36, 245-259

Blömer F, Günther H-O (2000): LP-based heuristics for scheduling chemical batch processes. International Journal of Production Research, 38, 1029-1051

Bogaschewsky R (1988): Dynamische Materialdisposition im Beschaffungsbereich. Simulation und Ergebnisanalyse. BME Schriftenreihe für das Materialmanagement, Frankfurt

Braun H (2002): Hierarchical planning in SAP supply chain management. Presentation at the GOR-AG „SCM-AG"-meeting in Darmstadt (18/10/02) URL (03/07/03): http://130.149.112.124:81/inhalt/scm-ag/Tagungen/2002_10_18/Vortraege/braun.pdf

Brown G, Keegan J, Vigus B, Wood K (2001): The Kellogg Company optimizes production, inventory, and distribution. Interfaces, 31, 6, 1-15

Brüggemann W, Jahnke H (1994): DLSP for two-stage multi-item batch production. International Journal of Production Research, 32, 755-768

Brüggemann W, Jahnke H (2000): The discrete lot-sizing and scheduling problem: complexity and modification for batch availability. European Journal of Operational Research, 124, 511-528

Carstensen P (2002): Simultane Losgrößen- und Reihenfolgeplanung bei konstantem Nachfrageverlauf. Deutscher Universitäts-Verlag (Gabler), Wiesbaden

Castro P, Barbosa-Póvoa APFD, Matos H (2001a): An improved RTN-continuous time formulation for the short-term scheduling of multipurpose batch plants. Industrial & Engineering Chemistry Research, 40, 2059-2068

Castro P, Barbosa-Póvoa APFD, Matos H (2001b): Reply to comments on "An improved RTN-continuous time formulation for the short-term scheduling of multipurpose batch plants". Industrial & Engineering Chemistry Research, 40, 5042

Castro PM, Barbosa-Póvoa AP, Matos HA (2003): Optimal periodic scheduling of batch plants using RTN-based discrete and continuous time formulations: a case study approach. Industrial & Engineering Chemistry Research, 42, 3346-3360

Castro PM, Barbosa-Póvoa AP, Matos HA, Novais AQ (2004): Simple continuous-time formulation for short-term scheduling of batch and continuous processes. Industrial & Engineering Chemistry Research, 43, 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, 477-486

Chase RB, Aquilano NJ, Jacobs FR (1998): Production and operations management: manufacturing and services. 8th edition, Irvin/McGraw-Hill, Boston

Chen C-L, Liu C-L, Feng X-D, Shao H-H (2002): Optimal short-term scheduling of multiproduct single-stage plants with parallel lines. Industrial & Engineering Chemistry Research, 41, 1249-1260

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

Constantino M (1998): Lower bound in lot-sizing models: a polyhedral study. Mathematics of Operations Research, 23, 101-118

Cordier C, Marchand H, Laundy R, Wolsey LA (1999): bc-opt: a branch-and-cut code for mixed integer programs. Mathematical Programming, Series A, 86, 335-353

Crama Y, Pochet Y, Wera Y (2001): A discussion of production planning approaches in the process industry. CORE Discussion Paper No. 42, Louvain-la-Neuve, Belgium

Dash Optimization (2002): Xpress-optimizer reference manual. Release 14. Blisworth, Northants, UK

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

Dennis D, Meredith J (2000): An empirical analysis of process industry transformation systems. Management Science, 46, 1085-1099

Derstroff MC (1995): Mehrstufige Losgrößenplanung mit Kapazitätsbeschränkungen. Physica-Verlag, Heidelberg

Dessouky MM, Kijowski BA (1997): Production scheduling of single-stage multi-product batch chemical processes with fixed batch sizes. IIE Transactions, 29, 399-408

Dillenberger C, Wollensak A (1994): Demand allocation for capacity optimization: a solution in IBM manufacturing. In: Ciriani, T.A., Leachman, R.C. (eds.): Optimization in industry 2, Mathematical Programming and Modeling Techniques in Practice. Wiley, Chichester, 89-98

Dillenberger C, Escudero LF, Wollensak A, Zhang W (1993): On solving a large-scale resource allocation problem in production planning. In: Fandel G, Gulledge T, Jones A (eds.): Operations research in production planning and control. Springer, Berlin et al., 105-119

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

Dixon PS, Silver EA (1981): A heuristic solution procedure for the multi-item, single level, limited capacity lot sizing problem. Journal of Operations Management, 2, 23-39

Dobson G (1987): The economic lot-scheduling problem: achieving feasibility using time-varying lot sizes. Operations Research, 35, 764-771

Domschke W, Scholl A (2000): Grundlagen der Betriebswirtschaftslehre. Springer, Berlin et al.

Domschke W, Scholl A, Voß S (1997): Produktionsplanung. Ablauforganisatorische Aspekte. 2nd edition, Springer, Berlin et al.

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

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

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

Eppen GD, Martin RK (1987): Solving multi-item capacitated lot-sizing problems using variable redefinition. Operations Research, 35, 832-848

Federgruen A, Tzur M (1994): Minimal forecast horizons and a new planning procedure for the general dynamic lot sizing model: nervousness revisited. Operations Research, 42, 456-468

Federgruen A, Meissner J, Tzur M (2003): Progressive interval heuristics for multi-item capacitated lot sizing problems. Working Paper, URL (13/08/04) http://www.meiss.com/download/SC-Federgruen-Meissner-Tzur.pdf

Fisher M, Ramdas K, Zheng Y-S (2001): Ending inventory valuation in multiperiod production scheduling. Management Science, 47, 679-692

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

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

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

Fleischmann B, Meyr H, Wagner M (2002): Advanced Planning. In: Stadtler H, Kilger C (eds.): Supply Chain Management and Advanced Planning. Concepts, Models, Software and Case Studies. 2nd edition, Springer, Berlin et al., 71-96

Florian M, Lenstra JK, Rinnooy Kan AHG (1980): Deterministic production planning: algorithms and complexity. Management Science, 26, 669-679

Giannelos NF, Georgiadis MC (2002a): A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes. Industrial & Engineering Chemistry Research, 41, 2178-2184

Giannelos NF, Georgiadis MC (2002b): A novel event-driven formulation for short-term scheduling of multipurpose continuous processes. Industrial & Engineering Chemistry Research, 41, 2431-2439

Gomory RE (1958): Outline of an algorithm for integer solutions to linear programs. Bulletin of the American Mathematical Society, 64, 275-278

Gopalakrishnan M (2000): A modified framework for modelling set-up carryover in the capacitated lotsizing problem. International Journal of Production Research, 38, 3421-3424

Gopalakrishnan M, Miller DM, Schmidt CP (1995): A framework for modelling setup carryover in the capacitated lotsizing problem. International Journal of Production Research, 33, 1973-1988

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

Grünert T (1998): Multi-level sequence-dependent dynamic lotsizing and scheduling. Shaker Verlag, Aachen

Grunow M, Günther H-O, Yang G (2003): Plant co-ordination in pharmaceutics supply networks. OR Spectrum, 25, 109-141

Günther H-O, Tempelmeier H (1997): Produktion und Logistik. 3rd edition, Springer, Berlin et al.

Gutenberg E (1983): Grundlagen der Betriebswirtschaftslehre, Band 1: Die Produktion. 24th edition, Springer, Berlin et al.

Haase K (1994): Lotsizing and scheduling for production planning. Lecture Notes in Economics and Mathematical System, No. 408, Springer, Berlin et al.

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

Haase K (1998): Capacitated lot-sizing with linked production quantities of adjacent periods. In: Drexl A, Kimms A (eds.): Beyond manufacturing resource planning (MRP II), Springer, Berlin et al., 127-146

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, 159-169

Harris FW (1913): How many parts to make at once. Factory, The magazine of management, 10, 135-136 and 152 (reprinted in: Operations Research (1990), 38, 947-950

Hauth M (1998): Hierarchisch integrierte Planungsansätze in der Prozeßfertigung. Deutscher Universitäts-Verlag (Gabler), Wiesbaden

Helber S (1994): Kapazitätsorientierte Losgrößenplanung in PPS-Systemen. M&P Verlag für Wissenschaft und Forschung, Stuttgart

Heuts RMJ, Seidel HP, Selen WJ (1992): A comparison of two lot sizing-sequencing heuristics for the process industry. European Journal of Operational Research, 59, 413-424

Hill AV (2003): The encyclopedia of operations management terms. Production Operations Management Society, Minneapolis

Hindi KS, Fleszar K, Charalambous C (2003): An effective heuristic for the CLSP with set-up times. Journal of the Operational Research Society, 54, 490-498

Hui C-W, Gupta A (2001): A bi-index continuous-time mixed-integer linear programming model for single-stage batch scheduling with parallel units. Industrial & Engineering Chemistry Research, 40, 5960-5967

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

Ierapetritou MG, Floudas CA (1998b): Effective continuous-time formulation for short-term scheduling. 2. Continuous and semicontinuous processes. Industrial & Engineering Chemistry Research, 37, 4360-4374

Ierapetritou MG, Floudas CA (2001): Comments on "An improved RTN-continuous time formulation for the short-term scheduling of multipurpose batch plants". Industrial & Engineering Chemistry Research, 40, 5040-5041

Ierapetritou MG, Hené TS, Floudas CA (1999): Effective continuous-time formulation for short-term scheduling. 3. Multiple intermediate due dates. Industrial & Engineering Chemistry Research, 38, 3446-3461

Johnson EL, Nemhauser GL, Savelsbergh MWP (2000): Progress in linear programming-based algorithms for integer programming: an exposition. INFORMS Journal on Computing, 12, 2-23

Jordan C (1996): Batching and scheduling. Models and methods for several problem classes. Lecture Notes in Economics and Mathematical Systems, No. 437, Springer, Berlin et al.

Jordan C, Drexl A (1998): Discrete lotsizing and scheduling by batch sequencing. Management Science, 44, 698-713

Kallrath J (1999): The concept of contiguity in models based on time-indexed formulations. In: Keil F, Mackens W, Werther J (eds.): Scientific computing in chemical engineering II. Springer, Berlin et al., 330-337

Kallrath J (2002a): Gemischt-ganzzahlige Optimierung: Modellierung in der Praxis. Vieweg, Braunschweig, Wiesbaden

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

Kallrath J, Wilson JM (1997): Business optimisation using mathematical programming. Macmillan Press, Houndmills, Basingstoke, London

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

Karimi IA, McDonald CM (1997): Planning and scheduling of parallel semicontinuous processes. 2. Short term scheduling. Industrial & Engineering Chemistry Research, 36, 2701-2714

Karimi B, Ghomi SMTF, Wilson JM (2003): The capacitated lot sizing problem: a review of models and algorithms. Omega, 31, 365-378

Karmarkar US, Schrage L (1985): The deterministic dynamic product cycling problem. Operations Research, 33, 326-345

Kelly JD (2002): Chronological decomposition heuristic for scheduling: divide and conquer method. AIChE Journal, 48, 2995-2999

Kießwetter M (1999): Ablaufplanung in der chemischen Industrie. Optimierung mit evolutionären Algorithmen. Deutscher Universitäts-Verlag (Gabler), Wiesbaden

Kimms A (1997): Multi-level lot sizing and scheduling. Physica-Verlag, Heidelberg

Kimms A, Drexl A (1998): Some insights into proportional lot sizing and scheduling. Journal of the Operational Research Society, 49, 1196-1205

Kimms A, Motta Toledo CF (2003): Bottling Coca-Cola Soft Drinks. Working Paper, Fakultät für Wirtschaftswissenschaften, TU Bergakademie Freiberg, Germany

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

Kuik R, Salomon M, van Wassenhove LN (1994): Batching decisions: structure and models. European Journal of Operational Research, 75, 243-263

Lamba N, Karimi IA (2002a): Scheduling parallel production lines with resource constraints. 1. Model formulation. Industrial & Engineering Chemistry Research, 41, 779-789

Lamba N, Karimi IA (2002b): Scheduling parallel production lines with resource constraints. 2. Decomposition algorithm. Industrial & Engineering Chemistry Research, 41, 790-800

Lambrecht MR, Vanderveken H (1979): Production scheduling and sequencing for multi-stage production systems. OR Spektrum, 1, 103-114

Lee YM, Chen EJ (2002): BASF uses a framework for developing web-based production-planning-optimization tools. Interfaces, 32, 6, 15-24

Lee K-H, Park HI, Lee I-B (2001): A novel nonuniform discrete time formulation for short term scheduling of batch and continuous processes. Industrial & Engineering Chemistry Research, 40, 4902-4911

Lee K-H, Heo S-K, Lee H-K, Lee I-B (2002): Scheduling of single-stage and continuous processes on parallel lines with intermediate due dates. Industrial & Engineering Chemistry Research, 41, 58-66

Leschke JP (1995): Plastech, Inc. – The importance of matching production capabilities and market requirements. Production and Inventory Management Journal, 36, Third Quarter, 11-15

Lim M-F, Karimi IA (2003a): A slot-based formulation for single-stage multiproduct batch plants with multiple orders per product. Industrial & Engineering Chemistry Research, 42, 1914-1924

Lim M-F, Karimi IA (2003b): Resource-constrained scheduling of parallel production lines using asynchronous slots. Industrial & Engineering Chemistry Research, 42, 6832-6842

Lin X, Floudas CA, Modi S, Juhasz NM (2002): Continuous-time optimization approach for medium-range production scheduling of a multiproduct batch plant. Industrial & Engineering Chemistry Research, 41, 3884-3906

Loos P (1997): Produktionslogistik in der chemischen Industrie: Betriebstypologische Merkmale und Informationsstrukturen. Gabler, Wiesbaden

Maes J, van Wassenhove LN (1988): Multi-item single-level capacitated lot-sizing heuristics: a general review. Journal of the Operational Research Society, 39, 991-1004

Maes J, McClain JO, van Wassenhove LN (1991): Multilevel capacitated lotsizing complexity and LP-based heuristics. European Journal of Operational Research, 53, 131-148

Magnanti TL, Vachani R (1990): A strong cutting plane algorithm for production scheduling with changeover costs. Operations Research, 38, 456-473

Magnusson T (2001): The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times. Master's thesis, University of Minnesota

Majozi T, Zhu FXX (2001): A novel continuous time MILP formulation for multipurpose batch plants. 1. Short term scheduling. Industrial & Engineering Chemistry Research, 40, 5935-5949

Maravelias CT, Grossmann IE (2003a): New general continuous time state-task network formulation for short-term scheduling of multipurpose batch plants. Industrial & Engineering Chemistry Research, 42, 3056-3074

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

Marchand H, Martin A, Weismantel R, Wolsey LA (2002): Cutting planes in integer and mixed integer programming. Discrete Applied Mathematics, 123, 397-446

McDonald CM, Karimi IA (1997): Planning and scheduling of parallel semicontinuous processes. 1. Production planning. Industrial & Engineering Chemistry Research, 36, 2691-2700

McKnew MA, Saydam C, Coleman BJ (1991): An efficient zero-one formulation of the multilevel lot-sizing problem. Decision Sciences, 22, 280-295

Meyr H (1999): Simultane Losgrößen- und Reihenfolgeplanung für kontinuierliche Produktionslinien – Modelle und Methoden im Rahmen des Supply Chain Management. Deutscher Universitäts-Verlag (Gabler), Wiesbaden

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

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

Meyr H (2004): Simultane Losgrößen- und Reihenfolgeplanung bei mehrstufiger kontinuierlicher Fertigung. Zeitschrift für Betriebswirtschaft, 74, 585-610

Meyr H, Wagner M, Rohde J (2002): Structure of Advanced Planning Systems. In: Stadtler H, Kilger C (eds.): Supply Chain Management and Advanced Planning. Concepts, Models, Software and Case Studies. 2nd edition, Springer, Berlin et al., 99-104

Miller CE, Tucker AW, Zemlin RA (1960): Integer programming formulation of traveling salesman problems. Journal of the ACM, 7, 326-329

Miller AJ, Nemhauser GL, Savelsbergh MWP (2000): Solving multi-item capacitated lot-sizing problems with setup times by branch-and-cut. CORE Discussion Papers No. 39, UCL, Louvain-la-Neuve, Belgium

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

Mockus L, Reklaitis GV (1999a): Continuous time representation approach to batch and continuous process scheduling. 1. MINLP formulation. Industrial & Engineering Chemistry Research, 38, 197-203

Mockus L, Reklaitis GV (1999b): Continuous time representation approach to batch and continuous process scheduling. 2. Computational issues. Industrial & Engineering Chemistry Research, 38, 204-210

Nemhauser GL, Wolsey LA (1988): Integer and combinatorial optimization. Wiley, New York

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

Overfeld J (1990): Produktionsplanung bei mehrstufiger Kampagnenfertigung. Peter Lang, Frankfurt

Pantelides CC (1994): Unified frameworks for optimal process planning and scheduling. In: Rippin DWT, Hale JC, Davis JF (eds.): Foundations of computer-aided process operations. CACHE, Austin, Texas, 253-274

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

Pekny JF, Reklaitis GV (1998): Towards the convergence of theory and practice: a technology guide for scheduling/planning methodology. In: Pekny JF, Blau GE, Carnahan B (eds.): Foundations of computer-aided process operations. AIChE Symposium Series, Vol. 94, CACHE, Snowbird, Utah, 91-111

Pinto JM, Grossmann IE (1998): Assignment and sequencing models for the scheduling of process systems. Annals of Operations Research, 81, 433-466

Plapp C (2003): Welchen Nutzen bringen MILP-basierte Planungsmodelle? PPS Management, 8, 4, 19-21

Pochet Y (2001): Mathematical programming models and formulations for deterministic production planning problems. In: Jünger M, Naddef D (eds.): Computational combinatorial optimization. Springer, Berlin et al., 57-111

Pochet Y, Wolsey LA (1994): Polyhedra for lot-sizing with Wagner-Whitin costs. Mathematical Programming, 67, 297-323

Pochet Y, Wolsey LA (1995): Algorithms and reformulations for lot sizing problems. In: Cook W, Lovasz L, Seymour P (eds.): Combinatorial Optimization. DIMACS Series in Discrete Mathematics and Computer Science, Vol. 20, AMS, Providence, 245-293

Porkka P (2000): Capacitated lot sizing problem with set-up time and set-up carry-over – Case Raflatac Oy. Master's thesis, Department of Marketing / Logistics, Helsinki School of Economics and Business Administration

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

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

Potts CN, van Wassenhove LN (1992): Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity. Journal of the Operational Research Society, 43, 395-406

Pressmar DB (1980): Modellierung und Optimierung dynamischer Produktionssysteme. In: Hahn D (ed.): Führungsprobleme industrieller Unternehmungen. Walter de Gruyter, Berlin, New York, 453-469

Quadt D, Kuhn H (2003): Production planning in semiconductor assembly. In: Papadopoulos CT (ed.): Proceedings of the Fourth Aegean International Conference on Analysis of Manufacturing Systems. Samos, Greece, 181-189

Quadt D, Kuhn H (2004): Capacitated lot-sizing and scheduling with parallel machines, back-orders and setup carry-over. Working paper, Catholic University of Eichstätt-Ingolstadt, Germany

Raflatac (2003), URL (13/08/04): http://www.raflatac.com/europe/eng/products/components/

Richter K (1975): Deterministische dynamische Mehrproduktmodelle der Produktionsplanung und Lagerhaltung. Mathematische Operationsforschung und Statistik, 6, 385-395

Rodrigues MTM, Latre LG, Rodrigues LCA (2000): Production planning using time windows for short-term multipurpose batch plants scheduling problems. Industrial & Engineering Chemistry Research, 39, 3823-3834

Rohde J (2004): Hierarchische Produktionsplanung bei losweiser Fertigung unter Einsatz künstlicher neuronaler Netze. PhD thesis, Technische Universität Darmstadt

Rohde J, Meyr H, Wagner M (2000): Die Supply Chain Planning Matrix. PPS Management, 5, 1, 10-15

Rosling K (1986): Optimal lot-sizing for dynamic assembly system. In: Axsäter S, Schneeweiß C, Silver E (eds.): Multi-stage production planning and inventory control. Springer, Berlin et al., 119-131

Salomon M (1991): Deterministic lotsizing models for production planning. Lecture Notes in Economics and Mathematical System, No. 355, Springer, Berlin et al.

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

Savelsbergh MWP (1994): Preprocessing and probing techniques for mixed integer programming problems. ORSA Journal on Computing, 6, 445-454

Schilling G, Pantelides CC (1996): A simple continuous-time process scheduling formulation and a novel solution algorithm. Computers & Chemical Engineering, 20, Supplement, S1221-S1226

Scholl A (2001): Robuste Planung und Optimierung: Grundlagen – Konzepte und Methoden – Experimentelle Untersuchungen. Physica-Verlag, Heidelberg

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

Selen WJ, Heuts RMJ (1990): Operational production planning in a chemical manufacturing environment. European Journal of Operational Research, 45, 38-46

Shah N (1998): Single- and multisite planning and scheduling: current status and future challenges. In: Pekny JF, Blau GE, Carnahan B (eds.): Foundations of computer-aided process operations. AIChE Symposium Series, Vol. 94, CACHE, Snowbird, Utah, 75-90

Shapiro JF (1993): Mathematical programming models and methods for production planning and scheduling. In: Graves SC, Rinnooy Kan AHG, Zipkin PH (eds.): Handbooks in OR and MS – Logistics of Production and Inventory. Vol. 4, North-Holland, Amsterdam et al., 371-443

Sherali HD, Smith JC (2001): Improving discrete model representations via symmetry considerations. Management Science, 47, 1396-1407

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

Smith-Daniels VL, Ritzman LP (1988): A model for lot sizing and sequencing in process industries. International Journal of Production Research, 26, 647-674

Smith-Daniels VL, Smith-Daniels DE (1986): A Mixed Integer Programming Model for Lot Sizing and Sequencing Packaging Lines in the Process Industries. IIE Transactions, 18, 278-285

Sox CR, Gao Y (1999): The capacitated lot sizing problem with setup carry-over. IIE Transactions, 31, 173-181

Stadtler H (1988): Hierarchische Produktionsplanung bei losweiser Fertigung. Physica-Verlag, Heidelberg

Stadtler H (1996): Mixed integer programming formulations for dynamic multi-item multi-level capacitated lotsizing. European Journal of Operational Research, 94, 561-581

Stadtler H (1997): Reformulations of the shortest route model for dynamic multi-item multi-level capacitated lotsizing. OR Spektrum, 19, 87-96

Stadtler H (2000): Improved rolling schedules for the dynamic single-level lot-sizing problem. Management Science, 46, 318-326

Stadtler H (2001): Losgrößenplanung: Von der Grundlagenforschung zur Anwendung im Supply Chain Management. Presentation at the GOR-AG „PPS"-

meeting in St. Leon-Rot (11/01/01) URL (13/08/04): http://130.149.112.124:81/PM/inhalt/de/forschung/scm-ag/2001_01_11/Stadtler.pdf

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

Stadtler H, Kilger C (2002) (eds.): Supply Chain Management and Advanced Planning. Concepts, Models, Software and Case Studies. 2nd edition, Springer, Berlin et al.

Staggemeier AT, Clark AR (2001): A survey of lot-sizing and scheduling models. In: Annals of the 23rd Annual Symposium (Simposio Brasileiro de Pesquisa Operacional) of the Brazilian Operational Research Society (SOBRAPO), November 6th-9th, 2001, 938-947

Stammen-Hegener C (2002): Simultane Losgrößen- und Reihenfolgeplanung bei ein- und mehrstufiger Fertigung. Deutscher Universitäts-Verlag (Gabler), Wiesbaden

Steven M (1994): Hierarchische Produktionsplanung. 2nd edition, Physica-Verlag, Heidelberg

Suerie C (2004): Campaign planning in time-indexed model formulations. International Journal of Production Research, accepted for publication

Suerie C, Stadtler H (2003): The capacitated lot-sizing problem with linked lot-sizes. Management Science, 49, 1039-1054

Tempelmeier H (2003): Material-Logistik. Modelle und Algorithmen für die Produktionsplanung und –steuerung und das Supply Chain Management. 5th edition, Springer, Berlin et al.

Tempelmeier H, Derstroff M (1996): A Lagrangean-based Heuristic for Dynamic Multilevel Multiitem Constrained Lotsizing with Setup Times. Management Science, 42, 738-757

Tempelmeier H, Helber S (1994): A heuristic for dynamic multi-item multi-level capacitated lotsizing for general product resources. European Journal of Operational Research, 75, 296-311

Thommen J-P (1991): Allgemeine Betriebswirtschaftslehre: umfassende Einführung aus managementorientierter Sicht. Gabler, Wiesbaden

Timpe C (2002): Solving planning and scheduling problems with combined integer and constraint programming. OR Spectrum, 24, 431-448

Timpe CH, Kallrath J (2000): Optimal planning in large multi-site production networks. European Journal of Operational Research, 126, 422-435

Trautmann N (2001): Anlagenbelegungsplanung in der Prozeßindustrie. Deutscher Universitäts-Verlag (Gabler), Wiesbaden

Trigeiro WW, Thomas LJ, McClain JO (1989): Capacitated lot sizing with setup times. Management Science, 35, 353-366

Trovinger SC, Bohn RE (2003): Setup time reduction for electronics assembly: combining simple (SMED) and sophisticated methods. Report 1997-03, The Information Storage Industry Center, University of California, San Diego

Vanderbeck F (1998): Lot-sizing with start-up times. Management Science, 44, 1409-1425

Voß S, Witt A (2003): Batching in der Produktionsplanung – Projektplanung mit reihenfolgeabhängigen Rüstkosten. Zeitschrift für Planung, 14, 75-89

Wagelmans A, van Hoesel S, Kolen A (1992): Economic lot sizing: an O(n log n) algorithm that runs in linear time in the Wagner-Whitin case. Operations Research, 40, Supplement No. 1, S145-S156

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

Wagner HM, Whitin TM (1959): Dynamic version of the economic lot size model. Management Science, 5, 89-96

Wang S, Guignard M (2002): Redefining event variables for efficient modeling of continuous-time batch processing. Annals of Operations Research, 116, 113-126

Webster S (1999): Remarks on: "Some Extensions of the Discrete Lotsizing and Scheduling Problem". Management Science, 45, 768-769

Wöhe G (1990): Einführung in die Allgemeine Betriebswirtschaftslehre. 17th edition, Vahlen, München

Wolsey LA (1998): Integer programming. Wiley, New York

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

Wolsey LA (2003): Strong formulations for mixed integer programs: valid inequalities and extended formulations. Mathematical Programming, Series B, 97, 423-447

Wu D, Ierapetritou MG (2003): Decomposition approaches for the efficient solution of short-term scheduling problems. Computers & Chemical Engineering, 27, 1261-1276

Xueya Z, Sargent RWH (1998): The optimal operation of mixed production facilities – extensions and improvements. Computers & Chemical Engineering, 22, 1287-1295

Yee KL, Shah N (1998): Improving the efficiency of discrete time scheduling formulation. Computers & Chemical Engineering, 22, Supplement, S403-S410

Zentner MG, Pekny JF, Reklaitis GV, Gupta JND (1994): Practical considerations in using model-based optimization for the scheduling and planning of batch/semicontinuous processes. Journal of Process Control, 4, 259-280

Zhang X, Sargent RWH (1996): The optimal operation of mixed production facilities – a general formulation and some approaches for the solution. Computers & Chemical Engineering, 20, 897-904

Zipkin PH (1991): Computing optimal lot sizes in the economic lot scheduling problem. Operations Research, 39, 56-63

Appendix – Model Formulations of Benchmarks

Gopalakrishnan (2000)

The benchmark model formulation by Gopalakrishnan (2000) consists of objective function (A-1) and constraints (A-2) – (A-17):

$$\text{Min} \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} h_{jt}\cdot I_{jt} + \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} sc_j\cdot Y_{jt} \tag{A-1}$$

$$I_{jt-1} + X_{jt} - I_{jt} = d_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \tag{A-2}$$

$$\sum_{j\in\mathcal{J}} a_j\cdot X_{jt} + \sum_{j\in\mathcal{J}} st_j\cdot Y_{jt} \le c_t \qquad \forall\, t\in\mathcal{T} \tag{A-3}$$

$$X_{jt} \le b_{jt}\cdot U_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \tag{A-4}$$

$$Y_{jt} = U_{jt} - S_{jt} + V_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \tag{A-5}$$

$$2\cdot S_{jt} \le \gamma_{jt-1} + \alpha_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T}\setminus\{1\} \tag{A-6}$$

$$V_{jt} \ge \gamma_{jt} - \beta_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \tag{A-7}$$

$$\omega_t \le \sum_{j\in\mathcal{J}} \alpha_{jt} \le 1 \qquad \forall\, t\in\mathcal{T} \tag{A-8}$$

$$\omega_t \le \sum_{j\in\mathcal{J}} \beta_{jt} \le 1 \qquad \forall\, t\in\mathcal{T} \tag{A-9}$$

$$\alpha_{jt} + \beta_{jt} \le 2 - \delta_t \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \tag{A-10}$$

$$\alpha_{jt} \le Y_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \tag{A-11}$$

$$\beta_{jt} \le Y_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \tag{A-12}$$

$$\sum_{j\in\mathcal{J}} \gamma_{jt} = 1 \qquad \forall\, t\in\mathcal{T} \tag{A-13}$$

$$Y_{jt} \le \omega_t \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \tag{A-14}$$

$$\sum_{j\in\mathcal{J}} Y_{jt} - 1 \le (J-1)\cdot\delta_t \qquad \forall\, t\in\mathcal{T} \tag{A-15}$$

$$I_{jt}, V_{jt}, X_{jt}, Y_{jt} \geq 0, \quad \alpha_{jt}, \beta_{jt}, \gamma_{jt}, S_{jt}, U_{jt} \in \{0;1\} \qquad \forall\, j \in \mathcal{J},\, t \in \mathcal{T} \quad \text{(A-16)}$$

$$\omega_t \geq 0, \quad 0 \leq \delta_t \leq 1 \qquad \forall\, t \in \mathcal{T} \quad \text{(A-17)}$$

Variables (if different from List of Symbols):

S_{jt} =1, if the setup for item *j* is carried into period *t*; =0 otherwise
U_{jt} =1, if item *j* is produced in period *t*; =0 otherwise
V_{jt} =1, if a setup operation for item *j* is performed at the end of period *t*; =0 otherwise
α_{jt} =1, if item *j* is produced first in period *t*; =0 otherwise
β_{jt} =1, if item *j* is produced last in period *t*; =0 otherwise
δ_t =0, if exactly one item is produced in period *t*; >0 otherwise
γ_{jt} =1, if the resource is set up for item *j* at the end of period *t*; =0 otherwise
ω_t =1, if at least one item is produced in period *t*; =0 otherwise

Remarks:

In contrast to the original model formulation by Gopalakrishnan (2000) no cost is charged in the objective function for production in a period, in which no setup operation occurs.

Ierapetritou et al. (1999)

The benchmark model formulation by Ierapetritou et al. (1999) consists of objective function (A-18) and constraints (A-19) – (A-51):

$$Min \quad \sum_{j \in \mathcal{J}\setminus\mathcal{J}_{out}} \sum_{\substack{t \in \mathcal{T} \\ \wedge e\ last\ in\ t}} h_{jt} \cdot I_{je-e(t)} + \sum_{j \in \mathcal{J}\setminus\mathcal{J}_{out}} \sum_{t \in \mathcal{T}} {h_{jt}}\big/_{2} \cdot \sum_{i \in I} \sum_{e \in \mathcal{T}_t} \sum_{m \in \mathcal{M}} X_{mie-1} \quad \text{(A-18)}$$

$$+ \sum_{i \in I} \sum_{l \in I} \sum_{\substack{m \in \mathcal{M} \\ \wedge i \in I_m \\ \wedge l \in I_m}} \sum_{e \in \mathcal{E}} sc_{il}^{sd} \cdot Y_{ile} + \sum_{m \in \mathcal{M}} \sum_{e \in \mathcal{E}} sc_{outage} \cdot Y_{me}$$

$$+ \sum_{j \in \mathcal{J}\setminus\mathcal{J}_{out}} \sum_{\substack{t \in \mathcal{T} \\ \wedge e\ last\ in\ t}} ssp_{jt} \cdot ISSV_{je} + \sum_{j \in \mathcal{J}\setminus\mathcal{J}_{out}} \sum_{\substack{t \in \mathcal{T} \\ \wedge e\ last\ in\ t}} blp_{jt} \cdot IB_{je}$$

$$\sum_{i \in I_m} V_{ie} + \sum_{o \in \mathcal{J}_{out}} V_{oe}^{out} = W_{me} \qquad \forall\, m \in \mathcal{M},\, e \in \mathcal{E} \quad \text{(A-19)}$$

$$W_{me} \leq 1 \qquad \forall\, m \in \mathcal{M},\, e \in \mathcal{E} \quad \text{(A-20)}$$

$$minrate_{mi} \cdot \left(PT_{mie}^{end} - PT_{mie}^{start}\right) \leq X_{mie} \qquad \forall\, m \in \mathcal{M},\, i \in I_m,\, e \in \mathcal{E} \quad \text{(A-21)}$$

$$maxrate_{mi} \cdot \left(PT_{mie}^{end} - PT_{mie}^{start}\right) \geq X_{mie} \qquad \forall\, m \in \mathcal{M},\, i \in I_m,\, e \in \mathcal{E} \quad \text{(A-22)}$$

$$I_{je} = I_{je-1} - S_{je} + \sum_{m\in\mathcal{M}} \sum_{\substack{i\in I_m \\ \wedge i\in\mathcal{J}_j}} X_{mie-1} \qquad \forall\, j\in\mathcal{J}\setminus\mathcal{J}_{out},\, e\in\mathcal{E} \quad \text{(A-23)}$$

$$IB_{je} = IB_{je-1} - S_{je} + d_{je} \qquad \forall\, j\in\mathcal{J}\setminus\mathcal{J}_{out},\, e\in\mathcal{E} \quad \text{(A-24)}$$

$$ISSV_{je} \geq sst_j - I_{je} \qquad \forall\, j\in\mathcal{J}\setminus\mathcal{J}_{out},\, e\in\mathcal{E} \quad \text{(A-25)}$$

$$PT^{end}_{mie} - PT^{start}_{mie} \leq \sum_{t\in\mathcal{T}} c_t \cdot V_{ie} \qquad \forall\, m\in\mathcal{M},\, i\in I_m,\, e\in\mathcal{E} \quad \text{(A-26)}$$

$$PT^{end}_{mie} - PT^{start}_{mie} \geq 0 \qquad \forall\, m\in\mathcal{M},\, i\in I_m,\, e\in\mathcal{E} \quad \text{(A-27)}$$

$$PT^{start}_{mie+1} \geq PT^{end}_{mle} \qquad \forall\, m\in\mathcal{M},\, i,l\in I_m,\, e\in\mathcal{E} \quad \text{(A-28)}$$

$$PT^{start}_{mie+1} \leq PT^{end}_{mle} + \sum_{t\in\mathcal{T}} c_t \cdot \left(2 - V_{ie+1} - V_{le}\right) \qquad \forall\, m\in\mathcal{M},\, i,l\in I_m,\, e\in\mathcal{E} \quad \text{(A-29)}$$

$$V_{ie} = V_{ie-1} \qquad \forall\, m\in\mathcal{M},\, i\in I_m,\, t\in\mathcal{T},\, e \text{ last in } t \quad \text{(A-30)}$$

$$Y_{ile} \leq V_{ie-1} \qquad \forall\, m\in\mathcal{M},\, i,l\in I_m,\, i\neq l,\, e\in\mathcal{E} \quad \text{(A-31)}$$

$$Y_{ile} \leq V_{le} \qquad \forall\, m\in\mathcal{M},\, i,l\in I_m,\, i\neq l,\, e\in\mathcal{E} \quad \text{(A-32)}$$

$$Y_{ile} \geq V_{ie-1} + V_{le} - 1 \qquad \forall\, m\in\mathcal{M},\, i,l\in I_m,\, i\neq l,\, e\in\mathcal{E} \quad \text{(A-33)}$$

$$Y_{me} \geq \sum_{i\in I_m} V_{ie} - \sum_{i\in I_m} V_{ie-1} \qquad \forall\, m\in\mathcal{M},\, e\in\mathcal{E} \quad \text{(A-34)}$$

$$Y_{me} \leq \sum_{i\in I_m} V_{ie} \qquad \forall\, m\in\mathcal{M},\, e\in\mathcal{E} \quad \text{(A-35)}$$

$$Y_{me} \leq 1 - \sum_{i\in I_m} V_{ie-1} \qquad \forall\, m\in\mathcal{M},\, e\in\mathcal{E} \quad \text{(A-36)}$$

$$\sum_{e\in\mathcal{E}_o} V^{out}_{oe} = 1 \qquad \forall\, o\in\mathcal{J}_{out} \quad \text{(A-37)}$$

$$PT^{end,out}_{moe} - PT^{start,out}_{moe} = minlot_{mo} \cdot V^{out}_{oe} \qquad \forall\, m\in\mathcal{M},\, o\in\mathcal{J}_{out},\, e\in\mathcal{E} \quad \text{(A-38)}$$

$$PT^{start}_{mie+1} \geq PT^{end,out}_{moe} \qquad \forall\, m\in\mathcal{M},\, i\in I_m,\, o\in\mathcal{J}_{out},\, e\in\mathcal{E} \quad \text{(A-39)}$$

$$PT^{end}_{mie} \leq PT^{start,out}_{moe+1} \qquad \forall\, m\in\mathcal{M},\, i\in I_m,\, o\in\mathcal{J}_{out},\, e\in\mathcal{E} \quad \text{(A-40)}$$

$$PT^{start,out}_{moe} \geq T^{out,start}_{o} \cdot V^{out}_{oe} \qquad \forall\, m\in\mathcal{M},\, o\in\mathcal{J}_{out},\, e\in\mathcal{E} \quad \text{(A-41)}$$

$$PT^{start,out}_{moe} \leq T^{out,end}_{o} - minlot_{mo} + \sum_{t\in\mathcal{T}} c_t \cdot \left(1 - V^{out}_{oe}\right) \qquad \forall\, m\in\mathcal{M},\, o\in\mathcal{J}_{out},\, e\in\mathcal{E} \quad \text{(A-42)}$$

$$\sum_{f=e-z+1}^{e}\left(PT_{mif}^{end} - PT_{mif}^{start}\right) \geq minlot_{mi} - \sum_{t\in\mathcal{T}} c_t \cdot \left(z - V_{ie+1} - V_{ie-z} - \sum_{f=e-z+1}^{e} V_{if}\right) \quad \forall\, m\in\mathcal{M},\, i\in I_m,\, z=1..Z,\, e=z\text{-}1..E\text{-}2 \quad \text{(A-43)}$$

$$PT_{mie}^{start} = \sum_{s=1}^{t} c_s \quad \forall\, m\in\mathcal{M},\, i\in I_m,\, t\in\mathcal{T},\, e \text{ last in } t \quad \text{(A-44)}$$

$$V_{ie} \in \{0,1\} \quad \forall\, i\in I,\, e\in\mathcal{E} \quad \text{(A-45)}$$

$$V_{oe}^{out} \in \{0,1\} \quad \forall\, o\in\mathcal{J}_{out},\, e\in\mathcal{E}_o \quad \text{(A-46)}$$

$$PT_{mie}^{end}, PT_{mie}^{start}, X_{mie} \geq 0 \quad \forall\, m\in\mathcal{M},\, i\in I_m,\, e\in\mathcal{E} \quad \text{(A-47)}$$

$$PT_{moe}^{end,out}, PT_{moe}^{start,out} \geq 0 \quad \forall\, o\in\mathcal{J}_{out},\, e\in\mathcal{E} \quad \text{(A-48)}$$

$$W_{me}, Y_{me} \geq 0 \quad \forall\, m\in\mathcal{M},\, e\in\mathcal{E} \quad \text{(A-49)}$$

$$Y_{ile} \geq 0 \quad \forall\, i,j\in I,\, e\in\mathcal{E} \quad \text{(A-50)}$$

$$I_{je}, IB_{je}, ISSV_{je}, S_{je} \geq 0 \quad \forall\, j\in\mathcal{J}\backslash\mathcal{J}_{out},\, e\in\mathcal{E} \quad \text{(A-51)}$$

Indices and index sets (if different from List of Symbols):

e	Event points ($e\in\mathcal{E}$)
i, l	Tasks ($i, l\in I$)
I	Set of tasks
I_m	Set of tasks producible on resource m
$\mathcal{J}_j$	Set of tasks producing product j
$\mathcal{J}_{out}$	Set of tasks representing outages
o	Tasks representing outages ($o\in\mathcal{J}_{out}$)
$\mathcal{T}_t$	Set of event points within period t

Data (if different from List of Symbols):

d_{je}	demand for product j at event point e
$e(t)$	Number of event points within period t
$maxrate_{mi}$	Maximal production rate of task i on resource m
$minrate_{mi}$	Minimal production rate of task i on resource m
sc_{outage}	Setup cost incurred after resource idle time or an outage
$T_o^{out,start}$	Earliest point in time, at which outage o is allowed to start
$T_o^{out,end}$	Latest point in time, at which outage o is allowed to end

Variables (if different from List of Symbols):

IB_{je}	Backlog of product j at event point e
$ISSV_{je}$	Safety stock violation of product j at event point e

PT^{end}_{mie}	Time that tasks i ends on resource m at event point e
$PT^{end,out}_{moe}$	Time that outage tasks o ends on resource m at event point e
PT^{start}_{mie}	Time that tasks i starts on resource m at event point e
$PT^{start,out}_{moe}$	Time that outage tasks o starts on resource m at event point e
S_{je}	Amount of product j shipped at event point e
V_{je}	Task start variable (=1, if task i starts at event point e, =0 otherwise)
V^{out}_{oe}	Outage task start variable (=1, if outage o starts at event point e, =0 otherwise)
W_{me}	Resource usage variable (=1, if resource m is utilized at event point e, =0 otherwise)
X_{mie}	Production of task i on resource m in the time slot associated with event point e
Y_{ile}	Setup variable (=1, if a setup operation from task i to task l is performed at event point e, =0 otherwise)
Y_{me}	Setup variable (=1, if a setup operation after idle time or after an outage task is performed on resource m at event point e, =0 otherwise)

Remarks:

Variables (resource-task-event point combinations) that do not exist or that are not allowed (in the case of outages) must not be defined or must be set to zero. In contrast to the original model formulation by Ierapetritou et al. (1999) the benchmark model formulation has been altered at several points. First, transition constraints ((27) – (31) in their paper) do not account for setup operations after idle time correctly as well as they do not allow transitions at three consecutive event points. Therefore, these constraints have been replaced by (A-31) – (A-36). Second, minimum run length constraints ((32) and (33) of their paper) force the minimum run length to be fulfilled within the first three event points, which is generalized to an arbitrary number Z+1 by (A-43). Here, Z is set to 4. Third, constraints (A-37) are added to force each outage to occur once. Last, to account for idle time correctly, constraints (25) of their paper are supplemented by (A-29) and (A-40).

Kallrath (1999)

The benchmark model formulation by Kallrath (1999) consists of objective function (2-20) and constraints (2-21), (2-22), (2-25) – (2-27), (6-1) – (6-4), (6-28) and (A-52) – (A-64):

$$Ct_{jt} = Ct_{jt-1} + Y_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \setminus \{1\} \quad \text{(A-52)}$$

$$Ct_{j1} = Y_{j1} \qquad \forall j \in \mathcal{J} \quad \text{(A-53)}$$

$$\sum_{n=0}^{N} Ca_{jtn} = 1 \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad \text{(A-54)}$$

$$\sum_{n=0}^{N} n \cdot Ca_{jtn} = Ct_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad \text{(A-55)}$$

$$Xa_{jtn} \le b_{jt} \cdot Ca_{jtn} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}, n \in \mathcal{N} \quad \text{(A-56)}$$

$$Xa_{jtn} \le X_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}, n \in \mathcal{N} \quad \text{(A-57)}$$

$$Xa_{jtn} \ge X_{jt} + b_{jt} \cdot Ca_{jtn} - b_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}, n \in \mathcal{N} \quad \text{(A-58)}$$

$$Xn_{jn} = \sum_{t=1}^{T} Xa_{jtn} \qquad \forall j \in \mathcal{J}, n \in \mathcal{N} \quad \text{(A-59)}$$

$$\{Ca_{jtn} \mid 0 \le n \le N\} \quad SOS1 \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \quad \text{(A-60)}$$

$$Xn_{jn} \in \{0; min\,lot_j..\infty\} \qquad \forall j \in \mathcal{J}, n \in \mathcal{N} \quad \text{(A-61)}$$

$$Xn_{jn} \le max\,lot_j \qquad \forall j \in \mathcal{J}, n \in \mathcal{N} \quad \text{(A-62)}$$

$$Xn_{jn} = bs_j \cdot Rn_{jn} \qquad \forall j \in \mathcal{J}, n \in \mathcal{N} \quad \text{(A-63)}$$

$$Rn_{jn} \ge 0 \quad \text{and integer} \qquad \forall j \in \mathcal{J}, n \in \mathcal{N} \quad \text{(A-64)}$$

Indices and index sets (if different from List of Symbols):

n Campaigns, $n \in \mathcal{N}$
$\mathcal{N}$ Set of (possible) campaigns (n=1..N)

Variables (if different from List of Symbols):

Ca_{jtn} Campaign activity (=1, if production of product j in period t belongs to campaign n, =0 otherwise)
Ct_{jt} Counting variable (= number of setups of product j in periods 1..t)
Rn_{jn} Batch variable (= number of batches produced in campaign n of product j)
Xa_{jtn} Production amount of product j in period t, which belongs to campaign n
Xn_{jn} Production amount of product j in campaign n

Remarks:

In contrast to the original model formulation by Kallrath (1999) the PLSP model formulation with valid inequalities as described in this thesis is used as a basis model (see section 7.2.1).

Karimi and McDonald (1997)

The benchmark model formulation by Karimi and McDonald (1997, model formulation M2) consists of objective function (A-65) and constraints (A-66) – (A-95):

$$Min \quad \sum_{j \in \mathcal{J} \setminus \mathcal{J}_{out}} \sum_{t \in \mathcal{T}} h_{jt} \cdot I_{jt-1} + \sum_{j \in \mathcal{J} \setminus \mathcal{J}_{out}} \sum_{t \in \mathcal{T}} {h_{jt}}/{2} \cdot X_{jt} \qquad \text{(A-65)}$$

$$+ \sum_{i \in \mathcal{J}} \sum_{\substack{j \in \mathcal{J} \\ \wedge j \neq i}} \sum_{\substack{m \in \mathcal{M} \\ \wedge j \in \mathcal{J}_m \\ \wedge i \in \mathcal{J}_m}} \sum_{s \in \mathcal{S}_m} sc_{mij}^{sd} \cdot Y_{mijs} + \sum_{j \in \mathcal{J} \setminus \mathcal{J}_{out}} \sum_{t \in \mathcal{T}} ssp_{jt} \cdot ISSV_{jt} + \sum_{j \in \mathcal{J} \setminus \mathcal{J}_{out}} \sum_{t \in \mathcal{T}} blp_{jt} \cdot IB_{jt}$$

$$\sum_{j \in \mathcal{J}_m} W_{mjs} = 1 \qquad \forall\, m \in \mathcal{M},\, s \in \mathcal{S}_m \qquad \text{(A-66)}$$

$$W_{mjs} = \sum_{i \in \mathcal{J}_m} Y_{mjis} \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m,\, s \in \mathcal{S}_m \qquad \text{(A-67)}$$

$$W_{mjs+1} = \sum_{i \in \mathcal{J}_m} Y_{mijs} \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m,\, s \in \mathcal{S}_m \qquad \text{(A-68)}$$

$$\sum_{\substack{m \in \mathcal{M} \\ \wedge j \in \mathcal{J}_m}} \sum_{s \in \mathcal{S}_{mt}} minrate_{mj} \cdot XT_{mjs} \leq X_{jt} \qquad \forall\, j \in \mathcal{J} \setminus \mathcal{J}_{out},\, t \in \mathcal{T} \qquad \text{(A-69)}$$

$$\sum_{\substack{m \in \mathcal{M} \\ \wedge j \in \mathcal{J}_m}} \sum_{s \in \mathcal{S}_{mt}} maxrate_{mj} \cdot XT_{mjs} \geq X_{jt} \qquad \forall\, j \in \mathcal{J} \setminus \mathcal{J}_{out},\, t \in \mathcal{T} \qquad \text{(A-70)}$$

$$PT_{ms} = PT_{ms-1} + \sum_{j \in \mathcal{J}_m} XT_{mjs} \qquad \forall\, m \in \mathcal{M},\, s \in \mathcal{S} \qquad \text{(A-71)}$$

$$Y_{mjjr-1} \geq W_{mjs} - Y_{mjjs-1} \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m,\, s \in \mathcal{S}_m,\, r \in \mathcal{S},\, s<r,\, r \leq g'(m,j,s),\; c_{t(s)} < minlot_{mj} \qquad \text{(A-72)}$$

$$Y_{mjjs-1} \geq W_{mj0} \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m,\, s \in \mathcal{S},\, s \leq g'(m,j,0),\, W_{mj0}=1 \qquad \text{(A-73)}$$

$$Y_{mjjs} \geq Y_{mjjs-1} \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m,\, t \in \mathcal{T},\, s \in \mathcal{S}_{mt},\, (s\text{-}1) \in \mathcal{S}_{mt},\, s \text{ not last in } t \qquad \text{(A-74)}$$

$$XT_{mjs} \leq c_t \cdot W_{mjs} \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m,\, t \in \mathcal{T},\, s \text{ first in } t \qquad \text{(A-75)}$$

$$XT_{mjs} \leq c_t \cdot \left(W_{mjs} - Y_{mjjs-1}\right) \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m,\, t \in \mathcal{T},\, s \text{ not first in } t \qquad \text{(A-76)}$$

$$XT_{mjs} + PT_{ms-1} \geq \overline{PT}_{ms} \cdot Y_{mjjs} \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m,\, s \in \mathcal{S}_m \qquad \text{(A-77)}$$

$$XT_{mjs} \geq minlot_{mj} \cdot \left(W_{mjs} - Y_{mjjs-1} - Y_{mjjs}\right) \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m,\, s \in \mathcal{S}_m,\; c_{t(s)} \geq minlot_{mj} \qquad \text{(A-78)}$$

$$XT_{mjs} + \sum_{\substack{p \in \mathcal{S} \\ s<p \leq r}} XT_{mjp} \geq minlot_{mj} \cdot \left(W_{mjs} - Y_{mjjs-1} - Y_{mjjr}\right) \qquad \forall\, m \in \mathcal{M}, j \in \mathcal{J}_m,\; s \in \mathcal{S}_m,\, r \in \mathcal{S},\, f(m,s) \leq r,\; r \leq g(m,j,s) \qquad \text{(A-79)}$$

$$I_{jt} = I_{jt-1} - S_{jt} + X_{jt} \qquad \forall\, j \in \mathcal{J} \setminus \mathcal{J}_{out},\, t \in \mathcal{T} \qquad \text{(A-80)}$$

$$ISSV_{jt} \geq sst_j - I_{jt} \qquad \forall\, j \in \mathcal{J} \setminus \mathcal{J}_{out},\, t \in \mathcal{T} \quad \text{(A-81)}$$

$$BL_{jt} = BL_{jt-1} - S_{jt} + d_{jt} \qquad \forall\, j \in \mathcal{J} \setminus \mathcal{J}_{out},\, t \in \mathcal{T} \quad \text{(A-82)}$$

$$\sum_{r=1}^{t} d_{jr} \geq \sum_{r=1}^{t} S_{jr} \qquad \forall\, j \in \mathcal{J} \setminus \mathcal{J}_{out},\, t \in \mathcal{T} \quad \text{(A-83)}$$

$$PT_{ms} \leq T_j^{out,end} \cdot W_{mjs} + \sum_{\substack{r \in \mathcal{T} \\ r \leq t}} c_r \cdot \left(1 - W_{mjs}\right) \qquad \forall\, m \in \mathcal{M},\, j \in \mathcal{J}_m \cap \mathcal{J}_{out},\, t \in \mathcal{T},\, s \in \mathcal{S}_{mt} \quad \text{(A-84)}$$

$$PT_{ms-1} \geq T_j^{out,start} \cdot W_{mjs} \qquad \forall\, m \in \mathcal{M},\, j \in \mathcal{J}_m \cap \mathcal{J}_{out},\, s \in \mathcal{S} \quad \text{(A-85)}$$

$$\sum_{s \in \mathcal{S}} \left(W_{mjs} - Y_{mjjs-1}\right) = 1 \qquad \forall\, m \in \mathcal{M},\, j \in \mathcal{J}_m \cap \mathcal{J}_{out} \quad \text{(A-86)}$$

$$\sum_{s \in \mathcal{S}} XT_{mjs} \geq minlot_{mj} \qquad \forall\, m \in \mathcal{M},\, j \in \mathcal{J}_m \cap \mathcal{J}_{out} \quad \text{(A-87)}$$

$$PT_{ms} \leq \overline{PT}_{ms} \qquad \forall\, m \in \mathcal{M},\, s \in \mathcal{S} \quad \text{(A-88)}$$

$$PT_{ms} \geq \underline{PT}_{ms} \qquad \forall\, m \in \mathcal{M},\, s \in \mathcal{S} \quad \text{(A-89)}$$

$$PT_{ms} = \sum_{\substack{r \in \mathcal{T} \\ r \leq t}} c_r \qquad \forall\, m \in \mathcal{M},\, t \in \mathcal{T},\, s \in \mathcal{S}_{mt},\, s \text{ last in } t \quad \text{(A-90)}$$

$$W_{mjs} \in \{0,1\} \qquad \forall\, m \in \mathcal{M},\, j \in \mathcal{J}_m,\, s \in \mathcal{S} \quad \text{(A-91)}$$

$$PT_{ms} \geq 0 \qquad \forall\, m \in \mathcal{M},\, s \in \mathcal{S} \quad \text{(A-92)}$$

$$XT_{mjs} \geq 0 \qquad \forall\, m \in \mathcal{M},\, j \in \mathcal{J}_m,\, s \in \mathcal{S} \quad \text{(A-93)}$$

$$Y_{mijs} \geq 0 \qquad \forall\, m \in \mathcal{M},\, i,j \in \mathcal{J}_m,\, s \in \mathcal{S} \quad \text{(A-94)}$$

$$I_{jt}, IB_{jt}, ISSV_{jt}, S_{jt}, X_{jt} \geq 0 \qquad \forall\, j \in \mathcal{J},\, t \in \mathcal{T} \quad \text{(A-95)}$$

Indices and index sets (if different from List of Symbols):

$\mathcal{J}_{out}$	Set of products representing outages
s	Slots ($s \in \mathcal{S}$)
$\mathcal{S}$	Set of slots
$\mathcal{S}_m$	Set of slots on resource *m*
$\mathcal{S}_{mt}$	Set of slots on resource *m* in period *t*

Data (if different from List of Symbols):

f(m,s)	First slot of the first period after slot *s* on resource *m*
g(m,j,s)	First slot of the first period after slot *s* on resource *m*, which will *guarantee* that the minimal campaign length of product *j* is fulfilled if the campaign has started in slot *s*

g'(m,j,s)	First slot of the first period after slot *s* on resource *m*, which must belong to the campaign of product *j* if the campaign has started in slot *s* in order to fulfill the minimal campaign length
$\overline{PT}_{ms}$	Latest point in time at which slots *s* ends on resource *m*
$\underline{PT}_{ms}$	Earliest point in time at which slots *s* ends on resource *m*
t(s)	Period to which slot *s* belongs
$T_j^{out,start}$	Earliest point in time at which outage *j* is allowed to start
$T_j^{out,end}$	Latest point in time at which outage *j* is allowed to end

Variables (if different from List of Symbols):

PT_{ms}	Point in time at which slot *s* ends (and *s*+1 begins) on resource *m*
S_{jt}	Amount of product *j* shipped at the end of period *t*
W_{mjs}	Production indicator variable (=1, if the setup state for product *j* persists on resource *m* in slot *s*, =0 otherwise)
XT_{mjs}	Production time for product *j* on resource *m* in slot *s*
Y_{mijs}	Setup variable (=1, if a setup operation from item *i* to item *j* is performed on resource *m* at the end of slot *s*, =0 otherwise)

Remarks:

Variables W_{mj0} and XT_{mj0} have to be initialized according to the data of the individual test instance. Variables (resource-product-period combinations) that do not exist or that are not allowed (in the case of outages) must not be defined or must be set to zero. In contrast to the original model formulation M2 by Karimi and McDonald (1997) the benchmark model formulation does not include the constraints that forbid idle time in a period, if backlog occurs in a later period, because the inclusion of this constraint will sometimes exclude the optimal solution (see also section 7.5.3).

Lee et al. (2002)

The benchmark model formulation by Lee et al. (2002) consists of objective function (A-96) and constraints (A-97) – (A-125):

$$Min \quad \sum_{j\in\mathcal{J}\setminus\mathcal{J}_{out}} \sum_{t\in\mathcal{T}} h_{jt}\cdot I_{jt-1} + \sum_{j\in\mathcal{J}\setminus\mathcal{J}_{out}} \sum_{t\in\mathcal{T}} \sum_{\substack{m\in\mathcal{M}\\ \wedge j\in\mathcal{J}_m}} {h_{jt}}/{2}\cdot X_{mjt} \tag{A-96}$$

$$+\sum_{i\in\mathcal{J}} \sum_{\substack{j\in\mathcal{J}\\ \wedge j\neq i}} \sum_{\substack{m\in\mathcal{M}\\ \wedge j\in\mathcal{J}_m\\ \wedge i\in\mathcal{J}_m}} \sum_{t\in\mathcal{T}_{mj}\cap\mathcal{T}_{mi}} sc^{sd}_{mij}\cdot Y_{mijt} + \sum_{j\in\mathcal{J}\setminus\mathcal{J}_{out}} \sum_{t\in\mathcal{T}} ssp_{jt}\cdot ISSV_{jt} + \sum_{j\in\mathcal{J}\setminus\mathcal{J}_{out}} \sum_{t\in\mathcal{T}} blp_{jt}\cdot IB_{jt}$$

$$Z_{mjt} \le U_{mjt} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m, t\in\mathcal{T}_{mj} \tag{A-97}$$

$$Z_{mjt-1} \le U_{mjt} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m, t\in\mathcal{T}_{mj} \tag{A-98}$$

$$Y_{mijt} + Y_{mjit} \le U_{mjt} \qquad \forall\ m\in\mathcal{M},\ i,j\in\mathcal{J}_m,\ i\neq j,\ t\in\mathcal{T}_{mj} \qquad \text{(A-99)}$$

$$\sum_{j\in\mathcal{J}_m}\ \sum_{\substack{i\in\mathcal{J}_m\\ \wedge i\neq j}} Y_{mijt} = \sum_{j\in\mathcal{J}_m} U_{mjt} - 1 \qquad \forall\ m\in\mathcal{M},\ t\in\mathcal{T} \qquad \text{(A-100)}$$

$$\sum_{j\in\mathcal{J}_m} U_{mjt} \ge 1 \qquad \forall\ m\in\mathcal{M},\ t\in\mathcal{T} \qquad \text{(A-101)}$$

$$Z_{mjt} = U_{mjt} - \sum_{\substack{i\in\mathcal{J}_m\\ \wedge i\neq j}} Y_{mjit} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m,\ t\in\mathcal{T}_{mj} \qquad \text{(A-102)}$$

$$\sum_{\substack{i\in\mathcal{J}_m\\ \wedge i\neq j}} Y_{mijt} \le 1 - Z_{mjt-1} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m,\ t\in\mathcal{T}_{mj} \qquad \text{(A-103)}$$

$$\sum_{j\in\mathcal{J}_m} XT_{mjt} = c_t \qquad \forall\ m\in\mathcal{M},\ t\in\mathcal{T} \qquad \text{(A-104)}$$

$$XT_{mjt} \le c_t \cdot U_{mjt} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m,\ t\in\mathcal{T}_{mj} \qquad \text{(A-105)}$$

$$XT_{mjt} \ge c_t - c_t \cdot \left(2 - Z_{mjt-1} - Z_{mjt}\right) \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m \setminus \mathcal{J}_{out},\ t\in\mathcal{T}_{mj} \qquad \text{(A-106)}$$

$$K_{mjt} \le \left(\sum_{s\in\mathcal{T}} c_s\right) \cdot Z_{mjt} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m \setminus \mathcal{J}_{out},\ t\in\mathcal{T}_{mj} \qquad \text{(A-107)}$$

$$K_{mjt} \le K_{mjt-1} + XT_{mjt} + \left(\sum_{s\in\mathcal{T}} c_s\right) \cdot \left(1 - Z_{mjt}\right) \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m \setminus \mathcal{J}_{out},\ t\in\mathcal{T}_{mj} \qquad \text{(A-108)}$$

$$K_{mjt-1} + XT_{mjt} \ge minlot_{mj} \cdot U_{mjt} - \left(\sum_{s\in\mathcal{T}} c_s\right) \cdot Z_{mjt} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m \setminus \mathcal{J}_{out},\ t\in\mathcal{T}_{mj} \qquad \text{(A-109)}$$

$$X_{mjt} \le maxrate_{mj} \cdot XT_{mjt} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m \setminus \mathcal{J}_{out},\ t\in\mathcal{T}_{mj} \qquad \text{(A-110)}$$

$$X_{mjt} \ge minrate_{mj} \cdot XT_{mjt} \qquad \forall\ m\in\mathcal{M}, j\in\mathcal{J}_m \setminus \mathcal{J}_{out},\ t\in\mathcal{T}_{mj} \qquad \text{(A-111)}$$

$$I_{jt} = I_{jt-1} - S_{jt} + \sum_{\substack{m\in\mathcal{M}\\ j\in\mathcal{J}_m}} X_{mjt} \qquad \forall\ j\in\mathcal{J} \setminus \mathcal{J}_{out},\ t\in\mathcal{T} \qquad \text{(A-112)}$$

$$ISSV_{jt} \ge sst_j - I_{jt} \qquad \forall\ j\in\mathcal{J} \setminus \mathcal{J}_{out},\ t\in\mathcal{T} \qquad \text{(A-113)}$$

$$BL_{jt} = BL_{jt-1} - S_{jt} + d_{jt} \qquad \forall\ j\in\mathcal{J} \setminus \mathcal{J}_{out},\ t\in\mathcal{T} \qquad \text{(A-114)}$$

$$\sum_{s=1}^{t} d_{js} \ge \sum_{s=1}^{t} S_{js} \qquad \forall\ j\in\mathcal{J} \setminus \mathcal{J}_{out},\ t\in\mathcal{T} \qquad \text{(A-115)}$$

$$\sum_{t\in\mathcal{T}_{mj}} U_{mjt} \ge 1 \qquad \forall\ m\in\mathcal{M}, j\in \mathcal{J}_m \cap \mathcal{J}_{out} \qquad \text{(A-116)}$$

$$\sum_{t\in\mathcal{T}_{mj}} U_{mjt} - \sum_{t\in\mathcal{T}_{mj}\setminus\{\mathrm{T}\}} Z_{mjt} = 1 \qquad \forall\ m\in\mathcal{M}, j\in \mathcal{J}_m \cap \mathcal{J}_{out} \qquad \text{(A-117)}$$

$$\sum_{t\in\mathcal{T}_{mj}} XT_{mjt} = minlot_{mj} \qquad \forall\ m\in\mathcal{M}, j\in \mathcal{J}_m \cap \mathcal{J}_{out} \qquad \text{(A-118)}$$

$$XT_{mjt} \le cao_{mjt} \qquad \forall\ m \in \mathcal{M}, j \in \mathcal{J}_m \cap \mathcal{J}_{out}, t \in \mathcal{T}_{mj} \quad \text{(A-119)}$$

$$\sum_{i \in \mathcal{J}_m} Y_{mij1} = 0 \qquad \forall\ m \in \mathcal{M}, j \in \mathcal{J}_{mt}, Z_{mj0}=1 \quad \text{(A-120)}$$

$$F_{mjt} \ge F_{mit} + 1 - (J+1) \cdot (1 - Y_{mjit}) \qquad \forall\ m \in \mathcal{M}, i,j \in \mathcal{J}_m, i \ne j, t \in \mathcal{T} \quad \text{(A-121)}$$

$$Y_{mijt} \in \{0,1\} \qquad \forall\ m \in \mathcal{M}, i,j \in \mathcal{J}_m, i \ne j, t \in \mathcal{T} \quad \text{(A-122)}$$

$$U_{mjt}, Z_{mjt} \in \{0,1\} \qquad \forall\ m \in \mathcal{M}, j \in \mathcal{J}_m, t \in \mathcal{T}_{mj} \quad \text{(A-123)}$$

$$F_{mjt}, K_{mjt}, X_{mjt}, XT_{mjt} \ge 0 \qquad \forall\ m \in \mathcal{M}, j \in \mathcal{J}_m, t \in \mathcal{T}_{mj} \quad \text{(A-124)}$$

$$I_{jt}, IB_{jt}, ISSV_{jt}, S_{jt} \ge 0 \qquad \forall\ j \in \mathcal{J}, t \in \mathcal{T} \quad \text{(A-125)}$$

Indices and index sets (if different from List of Symbols):

$\mathcal{J}_{out}$ Set of products representing outages
$\mathcal{T}_{mj}$ Set of periods, in which product j is producible on resource m

Data (if different from List of Symbols):

cao_{mjt} Capacity, which is maximally available on resource m for outage j in period t

Variables (if different from List of Symbols):

F_{mjt} Position variable (takes only integer values), the larger F_{mjt} the later product j is scheduled on resource m in period t
S_{jt} Amount of product j shipped at the end of period t
U_{mjt} Production indicator variable (=1, if the setup state for product j persists on resource m at some point in period t, =0 otherwise)

Remarks:

Variables K_{mj0} and Z_{mj0} have to be initialized according to the data of the individual test instance. Variables (resource-product-period combinations) that do not exist or that are not allowed (in the case of outages) must not be defined or must be set to zero. In contrast to the original model formulation by Lee et al. (2002) the benchmark model formulation is supplemented by sub-tour elimination constraints (A-121) to guarantee that the correct optimal solution is found (see also section 7.5.3).

Sox and Gao (1999)

The benchmark model formulation by Sox and Gao (1999) consists of objective function (A-126) and constraints (A-127) – (A-141):

$$\text{Min} \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} sc_j \cdot V_{jt} + \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\sum_{s=t+1}^{T+1}\sum_{\beta=0}^{1}\left[\left(sc_j + co_{jts}\right)\cdot X_{jt0s\beta} + co_{jts}\cdot X_{jt1s\beta}\right] \tag{A-126}$$

$$\sum_{j\in\mathcal{J}}\sum_{\alpha=0}^{1}\sum_{s=t+1}^{T+1}\sum_{\beta=0}^{1} a_j \cdot m_{jts} \cdot X_{jt\alpha s\beta} + \sum_{j\in\mathcal{J}} st_j \cdot Y_{jt} \le c_t \qquad \forall\, t\in\mathcal{T} \tag{A-127}$$

$$\sum_{t=1}^{td_j}\sum_{s=t+1}^{T+1}\sum_{\beta=0}^{1} X_{jt0s\beta} = 1 \qquad \forall\, j\in\mathcal{J} \tag{A-128}$$

$$\sum_{p=1}^{t-1}\sum_{\alpha=0}^{1} X_{jp\alpha t0} = \sum_{s=t+1}^{T+1}\sum_{\beta=0}^{1} X_{jt0s\beta} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T},\, t\ge td_j \tag{A-129}$$

$$\sum_{p=1}^{t-1}\sum_{\alpha=0}^{1} X_{jp\alpha t1} = \sum_{s=t+1}^{T+1}\sum_{\beta=0}^{1} X_{jt1s\beta} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T},\, t\ge td_j \tag{A-130}$$

$$\sum_{s=t+1}^{T+1}\sum_{\beta=0}^{1} X_{jt0s\beta} + V_{jt} = Y_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \tag{A-131}$$

$$\sum_{p=1}^{t-1}\sum_{\alpha=0}^{1}\sum_{s=t}^{T+1} X_{jp\alpha s1} \le W_{jt} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T}\setminus\{1\} \tag{A-132}$$

$$\sum_{j\in\mathcal{J}} W_{jt} \le 1 \qquad \forall\, t\in\mathcal{T}\setminus\{1\} \tag{A-133}$$

$$\sum_{p=1}^{t-1}\sum_{\alpha=0}^{1}\sum_{s=t+1}^{T+1} X_{jp\alpha s1} + Y_{jt} \le 1 \qquad \forall\, j,\, k\in\mathcal{J}, j\ne k,\, t\in\mathcal{T}\setminus\{1\} \tag{A-134}$$

$$W_{jt} \le Y_{jt-1} + W_{jt-1} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T}\setminus\{1\} \tag{A-135}$$

$$W_{jt} + W_{jt-1} - Y_{jt-1} + Y_{kt-1} \le 2 \qquad \forall\, j,\, k\in\mathcal{J}, j\ne k,\, t\in\mathcal{T}\setminus\{1\} \tag{A-136}$$

$$\sum_{j\in\mathcal{J}}\sum_{s=t}^{T+1}\sum_{\beta=0}^{1} X_{jt1s\beta} \le 1 \qquad \forall\, t\in\mathcal{T},\, t\le\max_{j\in\mathcal{J}}\{\, td_j \,\} \tag{A-137}$$

$$\sum_{s=1}^{T}\sum_{\alpha=0}^{1}\sum_{\beta=0}^{1} X_{js\alpha T+1\beta} = 1 \qquad \forall\, j\in\mathcal{J} \tag{A-138}$$

$$0 \le X_{jt\alpha s\beta} \le 1 \qquad \forall\, j\in\mathcal{J},\, t,\, s\in\mathcal{T},\, s\ge t,\, \alpha,\beta\in\{0,1\} \tag{A-139}$$

$$0 \le V_{jt} \le 1 \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \tag{A-140}$$

$$W_{jt}, Y_{jt} \in \{0,1\} \qquad \forall\, j\in\mathcal{J},\, t\in\mathcal{T} \tag{A-141}$$

Indices and index sets (if different from List of Symbols):

α, β — Arc index (=0, if a setup operation is incurred in the period; =1, if a setup carries over into the period)

Data (if different from List of Symbols):

co_{jts} Holding cost for flow variable $X_{jt\alpha s\beta}$ for product j from period t to period s: $co_{jts} = \sum_{p=t+1}^{s-1} \sum_{r=t}^{p-1} h_{jr} \cdot d_{jp}$

m_{jts} Cumulated demand of product j from period t to period $s-1$

td_j First period with positive demand of product j in the planning interval

Variables (if different from List of Symbols):

V_{jt} Setup variable (fraction of setup cost to be incurred in the objective function, which is not included in the flow variables $X_{jt\alpha s\beta}$

$X_{jt\alpha s\beta}$ Flow variable for product j from (period t; setup state α) to (period s; setup state β)

Remarks:

In contrast to the original model formulation by Sox and Gao (1999) the benchmark model formulation is extended to work also with setup times along the lines of Suerie and Stadtler (2003, pp. 1053-1054).

Lecture Notes in Economics and Mathematical Systems

For information about Vols. 1–459
please contact your bookseller or Springer-Verlag

Vol. 460: B. Fleischmann, J. A. E. E. van Nunen, M. G. Speranza, P. Stähly, Advances in Distribution Logistic. XI, 535 pages. 1998.

Vol. 461: U. Schmidt, Axiomatic Utility Theory under Risk. XV, 201 pages. 1998.

Vol. 462: L. von Auer, Dynamic Preferences, Choice Mechanisms, and Welfare. XII, 226 pages. 1998.

Vol. 463: G. Abraham-Frois (Ed.), Non-Linear Dynamics and Endogenous Cycles. VI, 204 pages. 1998.

Vol. 464: A. Aulin, The Impact of Science on Economic Growth and its Cycles. IX, 204 pages. 1998.

Vol. 465: T. J. Stewart, R. C. van den Honert (Eds.), Trends in Multicriteria Decision Making. X, 448 pages. 1998.

Vol. 466: A. Sadrieh, The Alternating Double Auction Market. VII, 350 pages. 1998.

Vol. 467: H. Hennig-Schmidt, Bargaining in a Video Experiment. Determinants of Boundedly Rational Behavior. XII, 221 pages. 1999.

Vol. 468: A. Ziegler, A Game Theory Analysis of Options. XIV, 145 pages. 1999.

Vol. 469: M. P. Vogel, Environmental Kuznets Curves. XIII, 197 pages. 1999.

Vol. 470: M. Ammann, Pricing Derivative Credit Risk. XII, 228 pages. 1999.

Vol. 471: N. H. M. Wilson (Ed.), Computer-Aided Transit Scheduling. XI, 444 pages. 1999.

Vol. 472: J.-R. Tyran, Money Illusion and Strategic Complementarity as Causes of Monetary Non-Neutrality. X, 228 pages. 1999.

Vol. 473: S. Helber, Performance Analysis of Flow Lines with Non-Linear Flow of Material. IX, 280 pages. 1999.

Vol. 474: U. Schwalbe, The Core of Economies with Asymmetric Information. IX, 141 pages. 1999.

Vol. 475: L. Kaas, Dynamic Macroeconomics with Imperfect Competition. XI, 155 pages. 1999.

Vol. 476: R. Demel, Fiscal Policy, Public Debt and the Term Structure of Interest Rates. X, 279 pages. 1999.

Vol. 477: M. Théra, R. Tichatschke (Eds.), Ill-posed Variational Problems and Regularization Techniques. VIII, 274 pages. 1999.

Vol. 478: S. Hartmann, Project Scheduling under Limited Resources. XII, 221 pages. 1999.

Vol. 479: L. v. Thadden, Money, Inflation, and Capital Formation. IX, 192 pages. 1999.

Vol. 480: M. Grazia Speranza, P. Stähly (Eds.), New Trends in Distribution Logistics. X, 336 pages. 1999.

Vol. 481: V. H. Nguyen, J. J. Strodiot, P. Tossings (Eds.). Optimation. IX, 498 pages. 2000.

Vol. 482: W. B. Zhang, A Theory of International Trade. XI, 192 pages. 2000.

Vol. 483: M. Königstein, Equity, Efficiency and Evolutionary Stability in Bargaining Games with Joint Production. XII, 197 pages. 2000.

Vol. 484: D. D. Gatti, M. Gallegati, A. Kirman, Interaction and Market Structure. VI, 298 pages. 2000.

Vol. 485: A. Garnaev, Search Games and Other Applications of Game Theory. VIII, 145 pages. 2000.

Vol. 486: M. Neugart, Nonlinear Labor Market Dynamics. X, 175 pages. 2000.

Vol. 487: Y. Y. Haimes, R. E. Steuer (Eds.), Research and Practice in Multiple Criteria Decision Making. XVII, 553 pages. 2000.

Vol. 488: B. Schmolck, Ommitted Variable Tests and Dynamic Specification. X, 144 pages. 2000.

Vol. 489: T. Steger, Transitional Dynamics and Economic Growth in Developing Countries. VIII, 151 pages. 2000.

Vol. 490: S. Minner, Strategic Safety Stocks in Supply Chains. XI, 214 pages. 2000.

Vol. 491: M. Ehrgott, Multicriteria Optimization. VIII, 242 pages. 2000.

Vol. 492: T. Phan Huy, Constraint Propagation in Flexible Manufacturing. IX, 258 pages. 2000.

Vol. 493: J. Zhu, Modular Pricing of Options. X, 170 pages. 2000.

Vol. 494: D. Franzen, Design of Master Agreements for OTC Derivatives. VIII, 175 pages. 2001.

Vol. 495: I. Konnov, Combined Relaxation Methods for Variational Inequalities. XI, 181 pages. 2001.

Vol. 496: P. Weiß, Unemployment in Open Economies. XII, 226 pages. 2001.

Vol. 497: J. Inkmann, Conditional Moment Estimation of Nonlinear Equation Systems. VIII, 214 pages. 2001.

Vol. 498: M. Reutter, A Macroeconomic Model of West German Unemployment. X, 125 pages. 2001.

Vol. 499: A. Casajus, Focal Points in Framed Games. XI, 131 pages. 2001.

Vol. 500: F. Nardini, Technical Progress and Economic Growth. XVII, 191 pages. 2001.

Vol. 501: M. Fleischmann, Quantitative Models for Reverse Logistics. XI, 181 pages. 2001.

Vol. 502: N. Hadjisavvas, J. E. Martínez-Legaz, J.-P. Penot (Eds.), Generalized Convexity and Generalized Monotonicity. IX, 410 pages. 2001.

Vol. 503: A. Kirman, J.-B. Zimmermann (Eds.), Economics with Heterogenous Interacting Agents. VII, 343 pages. 2001.

Vol. 504: P.-Y. Moix (Ed.), The Measurement of Market Risk. XI, 272 pages. 2001.

Vol. 505: S. Voß, J. R. Daduna (Eds.), Computer-Aided Scheduling of Public Transport. XI, 466 pages. 2001.

Vol. 506: B. P. Kellerhals, Financial Pricing Models in Continuous Time and Kalman Filtering. XIV, 247 pages. 2001.

Vol. 507: M. Koksalan, S. Zionts, Multiple Criteria Decision Making in the New Millenium. XII, 481 pages. 2001.

Vol. 508: K. Neumann, C. Schwindt, J. Zimmermann, Project Scheduling with Time Windows and Scarce Resources. XI, 335 pages. 2002.

Vol. 509: D. Hornung, Investment, R&D, and Long-Run Growth. XVI, 194 pages. 2002.

Vol. 510: A. S. Tangian, Constructing and Applying Objective Functions. XII, 582 pages. 2002.

Vol. 511: M. Külpmann, Stock Market Overreaction and Fundamental Valuation. IX, 198 pages. 2002.

Vol. 512: W.-B. Zhang, An Economic Theory of Cities.XI, 220 pages. 2002.

Vol. 513: K. Marti, Stochastic Optimization Techniques. VIII, 364 pages. 2002.

Vol. 514: S. Wang, Y. Xia, Portfolio and Asset Pricing. XII, 200 pages. 2002.

Vol. 515: G. Heisig, Planning Stability in Material Requirements Planning System. XII, 264 pages. 2002.

Vol. 516: B. Schmid, Pricing Credit Linked Financial Instruments. X, 246 pages. 2002.

Vol. 517: H. I. Meinhardt, Cooperative Decision Making in Common Pool Situations. VIII, 205 pages. 2002.

Vol. 518: S. Napel, Bilateral Bargaining. VIII, 188 pages. 2002.

Vol. 519: A. Klose, G. Speranza, L. N. Van Wassenhove (Eds.), Quantitative Approaches to Distribution Logistics and Supply Chain Management. XIII, 421 pages. 2002.

Vol. 520: B. Glaser, Efficiency versus Sustainability in Dynamic Decision Making. IX, 252 pages. 2002.

Vol. 521: R. Cowan, N. Jonard (Eds.), Heterogenous Agents, Interactions and Economic Performance. XIV, 339 pages. 2003.

Vol. 522: C. Neff, Corporate Finance, Innovation, and Strategic Competition. IX, 218 pages. 2003.

Vol. 523: W.-B. Zhang, A Theory of Interregional Dynamics. XI, 231 pages. 2003.

Vol. 524: M. Frölich, Programme Evaluation and Treatment Choise. VIII, 191 pages. 2003.

Vol. 525:S. Spinler, Capacity Reservation for Capital-Intensive Technologies. XVI, 139 pages. 2003.

Vol. 526: C. F. Daganzo, A Theory of Supply Chains. VIII, 123 pages. 2003.

Vol. 527: C. E. Metz, Information Dissemination in Currency Crises. XI, 231 pages. 2003.

Vol. 528: R. Stolletz, Performance Analysis and Optimization of Inbound Call Centers. X, 219 pages. 2003.

Vol. 529: W. Krabs, S. W. Pickl, Analysis, Controllability and Optimization of Time-Discrete Systems and Dynamical Games. XII, 187 pages. 2003.

Vol. 530: R. Wapler, Unemployment, Market Structure and Growth. XXVII, 207 pages. 2003.

Vol. 531: M. Gallegati, A. Kirman, M. Marsili (Eds.), The Complex Dynamics of Economic Interaction. XV, 402 pages, 2004.

Vol. 532: K. Marti, Y. Ermoliev, G. Pflug (Eds.), Dynamic Stochastic Optimization. VIII, 336 pages. 2004.

Vol. 533: G. Dudek, Collaborative Planning in Supply Chains. X, 234 pages. 2004.

Vol. 534: M. Runkel, Environmental and Resource Policy for Consumer Durables. X, 197 pages. 2004.

Vol. 535: X. Gandibleux, M. Sevaux, K. Sörensen, V. T'kindt (Eds.), Metaheuristics for Multiobjective Optimisation. IX, 249 pages. 2004.

Vol. 536: R. Brüggemann, Model Reduction Methods for Vector Autoregressive Processes. X, 218 pages. 2004.

Vol. 537: A. Esser, Pricing in (In)Complete Markets. XI, 122 pages, 2004.

Vol. 538: S. Kokot, The Econometrics of Sequential Trade Models. XI, 193 pages. 2004.

Vol. 539: N. Hautsch, Modelling Irregularly Spaced Financial Data. XII, 291 pages. 2004.

Vol. 540: H. Kraft, Optimal Portfolios with Stochastic Interest Rates and Defaultable Assets. X, 173 pages. 2004.

Vol. 541: G.-Y. Chen, Vector Optimization (planned).

Vol. 542: J. Lingens, Union Wage Bargaining and Economic Growth. XIII, 199 pages. 2004.

Vol. 543: C. Benkert, Default Risk in Bond and Credit Derivatives Markets. IX, 135 pages. 2004.

Vol. 544: B. Fleischmann, A. Klose, Distribution Logistics. X, 284 pages. 2004.

Vol. 545: R. Hafner, Stochastic Implied Volatility. XI, 229 pages. 2004.

Vol. 546: D. Quadt, Lot-Sizing and Scheduling for Flexible Flow Lines. XVIII, 227 pages. 2004.

Vol. 547: M. Wildi, Signal Extraction. XI, 279 pages. 2005.

Vol. 548: D. Kuhn, Generalized Bounds for Convex Multistage Stochastic Programs. XI, 190 pages. 2005.

Vol. 549: G. N. Krieg, Kanban-Controlled Manufacturing Systems. IX, 236 pages. 2005.

Vol. 550: T. Lux, S. Reitz, E. Samanidou, Nonlinear Dynamics and Heterogeneous Interacting Agents. XIII, 327 pages. 2005.

Vol. 551: J. Leskow, M. Puchet Anyul, L. F. Punzo, New Tools for Economic Dynamics. XIX, 392 pages. 2005.

Vol. 552: C. Suerie, Time Continuity in Discrete Time Models. XVIII, 229 pages. 2005.