

Modeling lotsizing and scheduling problems with sequence dependent setups

Luis Guimarães^a, Diego Klabjan^b, Bernardo Almada-Lobo^a

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

^b*Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA*

Abstract

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

Keywords: Production Planning, Lotsizing and Scheduling, Mixed-Integer Programming, Sequence-dependent Setups, Non-triangular Setups, Computational Benchmark

1. Introduction

Several companies face the problem of timing and sizing production lots over a given planning horizon. Additionally, in many of these production environments, switching between production lots of two different products triggers operations, such as machine adjustments and cleansing procedures. These setup operations, which are dependent on the sequence, consume scarce production time and may cause additional costs due to, for example, losses in raw materials or intermediate products. Consequently, the production sequence must be explicitly embedded in the lot definition and scheduling. Lot sizing determines the timing and level of production to satisfy deterministic product demand over a finite planning horizon. Sequencing establishes the order in which lots are executed within a time period, accounting for the sequence-dependent setup times and costs. Integration of these two problems enables the creation of better production plans than those obtained when solving the two problems hierarchically by inducing the solution of the lotsizing problem in the scheduling level. Production plans are created with the objective of minimizing the overall costs consisting mainly of stock holding and setups, while satisfying the available capacity in each time period from which the expenditure in setup times is deducted.

This production scenario is present in many process industries, in which an efficient use of the available capacity is key to stay competitive in the current market environment. In the beverage industry sequence dependent

Email addresses: guimaraes.luis@fe.up.pt (Luis Guimarães), d-klabjan@northwestern.edu (Diego Klabjan), almada.lobos@fe.up.pt (Bernardo Almada-Lobo)

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

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

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

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

solution of a series of sub-MIPs, also depend on the generation of good upper bounds. Hence, this assessment of the formulations can potentially contribute to the identification of the potentially most efficient MIP formulations to be used in these hybrid methods.

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

The remainder of this paper is organized as follows. Section 2 presents the proposed classification framework for the modeling approaches. In Section 3 we describe the problem under study and all the assumptions made. Following the classes defined in our framework Sections 4 and 5 present the reviewed models, as well as introduce the new formulation proposed herein. Computational experiments assessing the models performance are shown in Section 6. Finally, Section 7 is devoted to final remarks, where conclusions from this work and some potential future research directions are highlighted.

2. Modeling sequence-dependent setups

In this section, we introduce a framework to classify the discrete time modeling approaches existing in the literature for LS with sequencing decisions. The framework is organized along two main sequencing dimensions: technique and time structure (see Figure 1). A class is defined by the technique and time structure used, e.g. product oriented macro period (PO-MP) models.

The sequence of production lots in a machine can be categorized following the definitions given by Kang et al. (1999): a *production-sequence* refers to the sequence of products being produced on the machine over the entire planning horizon and a *period-sequence* denotes the sequence of setup states within a time period. In discrete time models for LS with sequencing decisions a production-sequence decomposes into period-sequences, hence the term sequence will be used hereafter to refer to period-sequences. The first dimension used for classification regards the technique used to capture sequencing decisions. Two main approaches are distinguished: product oriented (PO) and sequence oriented (SO) formulations. When using a PO technique, sequences are explicitly defined by the MIP model, while in SO formulations the MIP model prescribes for each period a sequence from a pre-determined set of sequences, i.e. the model selects one sequence from the set.

Consider the representations of sequences depicted in Figure 2. By definition a sequence is a connected directed graph where each node i represents a production lot of product i and arc (i, j) indicates a setup from product i to product j . Additionally, the dashed arcs identify the first (input arc) and the last (output arc) production lots in the sequence, i.e. the initial and final setup state of the machine. A SO formulation corresponds to the selection of a connected graph (sequence) to be applied in each time period, thus it does not require additional constraints to ensure the connectivity of the setup decisions. On the other hand, a PO formulation operates on the selection of arcs (setups) to be performed in each time period, hence the so-called disconnected subtour elimination constraints,

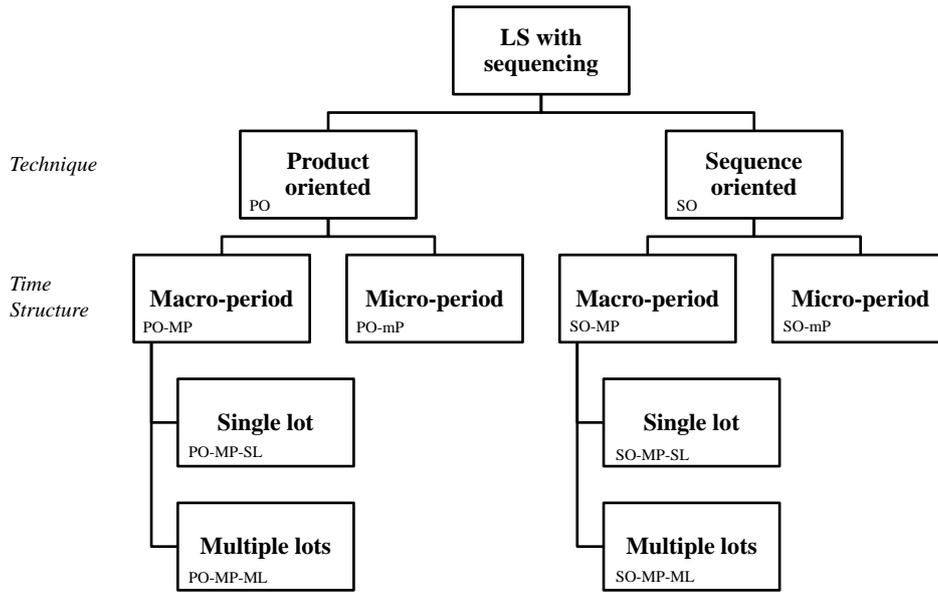


Figure 1: Proposed classification framework

which can be of an exponential size, are often required to ensure the connectivity of the subgraph induced by setup decisions. This is a major difference between these two approaches and explains why sequence oriented based formulations are easier to model. However, this potential advantage has the drawback of the number of possible sequences (decision variables) growing exponentially with the number of products present in the problem instance.

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

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

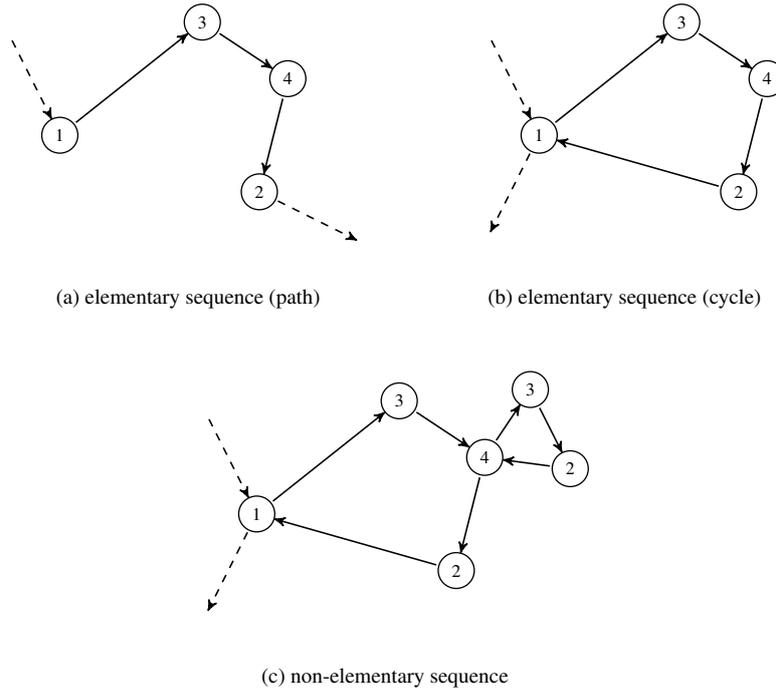


Figure 2: Examples of possible sequences.

(4-2) to establish the sequence depicted in the example.

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

MP models can be further divided according to the number of production lots of each product allowed to start within a time period into single lot (SL) and multiple lots (ML) models, giving origin to subclasses. Usually, setups obey the triangle inequality with respect to both the setup time and costs, i.e. it is more efficient to change directly between two products than via a third product. Under this setting in any optimal solution, at most one setup for each product per time period is performed (single lot). Nevertheless, in some industries, contamination occurs when changing from one product to another implying additional cleansing operations. If a ‘cleansing’ or shortcut product can absorb contamination while being produced and therefore replacing the cleansing operations, non-triangular setups appear. Thus, allowing multiple lots of each product per time period can potentially reduce setup times and costs. The need for multiple lots can also come from industries where production batches are

bounded or of fixed size. This distinction is made since tackling multiple lots in the same period is a non trivial extension to most models. Furthermore, minimum lot size is important in the case of non-triangular setups to avoid fictitious setups via empty product lots (zero production). Figures 2a and 2b are examples of sequences which can be obtained by using MP-SL models, while the sequence illustrated in Figure 2c can only be achieved using a MP-ML formulation.

Table 1 presents the models which will be reviewed in the following sections in each of the classes defined.

Table 1: Classification of lotsizing and scheduling models with sequencing decisions

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

3. Problem definition

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

- d_{it} demand of product i in period t (units),
- h_{it} holding cost of one stock unit of product i in period t (cost/unit),
- cap_t capacity of the machine in period t (time),
- p_{it} processing time of product i in period t (time/unit),
- b_{it} the maximum amount of product i that can be produced in period t (units),
- sc_{ij} cost incurred to set up the machine from product i to product j (cost),
- st_{ij} time needed to set up the machine from product i to product j (time),
- m_i minimum lot size of product i (units).

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

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

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

4. Product oriented formulations

4.1 Micro-period models

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

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

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

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

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

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

$$\text{GLSP} \quad \min \sum_{i,t} h_{it} \cdot I_{it} + \sum_{i,j,n} sc_{ij} \cdot T_{ijn} \quad (1)$$

$$\text{s.t.} \quad I_{i,t-1} + \sum_{n \in A_t} X_{in} = d_{it} + I_{it} \quad \forall i, t, \quad (2)$$

$$\sum_{i,n \in A_t} p_{it} \cdot X_{in} + \sum_{i,j,n \in A_t} st_{ij} \cdot T_{ijn} \leq cap_t \quad \forall t, \quad (3)$$

$$X_{in} \leq b_{it} \cdot Y_{in} \quad \forall i, t, n \in A_t, \quad (4)$$

$$\sum_i Y_{in} = 1 \quad \forall n, \quad (5)$$

$$T_{ijn} \geq Y_{i,n-1} + Y_{jn} - 1 \quad \forall i, j, n, \quad (6)$$

$$X_{in} \geq m_i \cdot (Y_{in} - Y_{i,n-1}) \quad \forall i, n, \quad (7)$$

$$X, I, T \geq 0, \quad Y \in \{0, 1\}. \quad (8)$$

The objective function (1) minimizes the total sum of holding and setup costs. Inventory balance constraints (2) satisfy demand either from initial inventory or production within the current period. The total period's production of each product is obtained by summing up the productions in the different micro-periods. Inequalities (3) ensure that the total production time plus the required setup time does not exceed the available capacity. The correct relation between production quantities and the machine setup state in each micro-period is expressed by (4), while (5) enforce a single setup state per micro-period. Constraints (6) trace changeovers throughout the planning horizon. Minimum lot sizes are introduced by constraints (7) to prevent empty lots and thus an incorrect evaluation of setup times and cost if the setup matrix does not obey the triangle inequality.

Figure 3 graphically represents the same sequence depicted in Figure 2c using the network interpretation of the GLSP and dividing a macro-period into 8 micro-periods. Essentially, it corresponds to a path in a directed graph where nodes are possible setup states in each micro-period and arcs connecting setup states are changeovers.

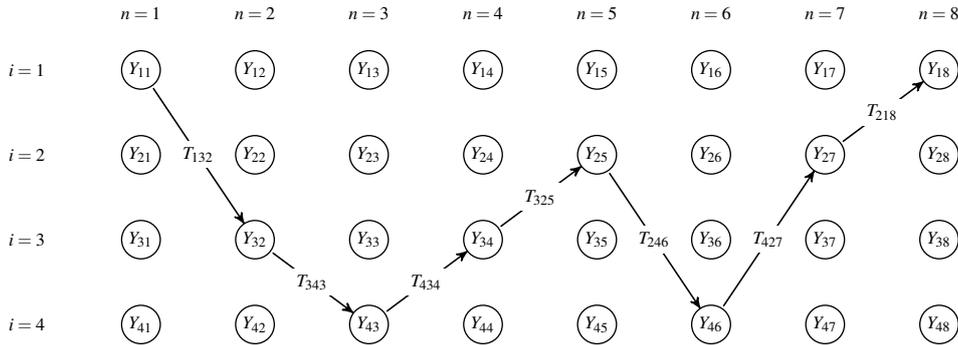


Figure 3: An example of sequence $\{1-3-4-3-2-4-2-1\}$ using a PO-mP formulation

As suggested by the previous figure, changeover constraints (6) can be reformulated as a shortest path or network flow (NF) problem from the first micro-period to the last. This yields a substantially tighter model as shown in Wolsey (1997). The strength of the (GLSP^{NF}) reformulated model comes from the fact that, when no

other constraints are present, the extreme points of the relaxed problem are integer solutions. The reformulated constraints are stated as:

$$\sum_j T_{ijn} = Y_{i,n-1} \quad \forall i, n, \quad (9)$$

$$\sum_i T_{ijn} = Y_{jn} \quad \forall j, n. \quad (10)$$

Constraints (9) force a changeover from the product produced in the previous micro-period at the beginning of the current micro-period. Similarly, constraints (10) force a changeover to the product being produced in each micro-period. Note that it is possible to have a changeover from product i to itself and thus these constraints act as setup state conservation constraints allowing production lots to span over multiple micro-periods. The complete GLSP^{NF} model reads (1) - (5) and (7) - (10).

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

$$\text{CC} \quad \min(1)$$

$$\text{s.t.} \quad (2) - (3)$$

$$\sum_{i,j} T_{ji0} = 1 \quad (11)$$

$$X_{in} \leq b_{it} \cdot \sum_j T_{jin} \quad \forall i, t, n \in A_t, \quad (12)$$

$$\sum_j T_{ji,n-1} = \sum_j T_{ijn} \quad \forall i, n, \quad (13)$$

$$X_{in} \geq m_i \cdot \sum_{j \neq i} T_{jin} \quad \forall i, n, \quad (14)$$

$$X, I \geq 0, \quad T \in \{0, 1\}. \quad (15)$$

In this formulation, constraint (11) defines the initial setup state of the machine. Constraints (12) guarantee that production of a given product only occurs if the machine is set up at the beginning of the micro-period, which can be either by an actual setup or via conservation of the previous setup state. Flow constraints (13) simultaneously keep track of changeovers and machine configuration state. These constraints have a similar structure as constraints (9) - (10) also capturing the network flow interpretation of the model. Minimum lotsizes (14) are again imposed to address non-triangular setups. In Appendix Appendix A we show that the CC formulation is stronger than the original GLSP formulation.

4.2 Macro-period models

The problem of extending the traditional capacitated lot sizing problem (CLSP) to account for sequencing decisions is known as the CLSD. To formulate the CLSD we introduce the binary decision variables Z_{it} which equals one if the machine is set up for product i at the beginning of period t capturing the setup state conservation among adjacent periods. We also update the definition of variables T_{ijt} to be the number of changeovers from

product i to product j in time period t and include q_{it} as an upper bound on the number of setups to product i in period t . A general CLSD model formulation reads:

$$\text{CLSD} \quad \min \sum_{i,t} h_{it} \cdot I_{it} + \sum_{i,j,t} sc_{ij} \cdot T_{ijt} \quad (16)$$

$$\text{s.t.} \quad I_{i,t-1} + X_{it} = d_{it} + I_{it} \quad \forall i, t, \quad (17)$$

$$\sum_i p_{it} \cdot X_{it} + \sum_{i,j} st_{ij} \cdot T_{ijt} \leq cap_t \quad \forall t, \quad (18)$$

$$X_{it} \leq b_{it} \cdot \left(\sum_j T_{jit} + Z_{it} \right) \quad \forall i, t, \quad (19)$$

$$\sum_i Z_{it} = 1 \quad \forall t, \quad (20)$$

$$Z_{it} + \sum_j T_{jit} = \sum_j T_{ijt} + Z_{i,t+1} \quad \forall i, t, \quad (21)$$

$$X, I \geq 0, \quad Z \in \{0, 1\}, \quad T_{ijt} \in \{0, \dots, q_{jt}\}, \quad (22)$$

$$\{(i, j) : T_{ijt} > 0\} \text{ does not include disconnected subtours } \forall t. \quad (23)$$

As in mP models the objective function (16) minimizes the total expenditure in holding and setup costs. Constraints (17) and (18) express, respectively, the common inventory balance and capacity constraints from large bucket formulations. Production is linked by the machine setup state through constraints (19); production may only occur if a setup is carried over from the previous period or at least one setup is performed in the period. Constraints (20) guarantee that the machine is set up for a single product in the beginning of each time period. Machine configuration is traced in (21) which ensures a balanced flow of setups. If there are no setups in period t the machine configuration is carried to period $t + 1$. On the other hand, for each product i three cases may appear: more input than output setups, more output than input setups and equal number of input and output setups. The first case forces the machine to be set up for product i in the beginning of the next period. Similarly, in the second case the machine must be set up for product i in the beginning of period t . Finally, the third case happens when the product is neither the first nor the last in the sequence, or no setup occurs in the period.

The formulation would be incomplete without constraints (23) which prevent disconnected subtours to create feasible integer solutions. Subtours are sequences that start and end at the same setup state. Figure 4 illustrates examples of subtours that can appear in a solution for the CLSD without constraints (23) and classified according to the notation of Menezes et al. (2011). An alpha subtour (see Figure 4a) is defined by $Z_{it} = Z_{i,t+1} = 1$ and $\sum_{i,j} T_{ijt} \geq 1$, meaning that at least one setup is performed in time period t and the machine configuration at the beginning and end of the period is the same. Figure 4b depicts the case of a connected subtour created by the existence of two production lots for product 2. Figures 4c and 4d show examples of disconnect subtours, i.e. sequences that are not connected to the main sequence. This class of subtours can be divided into simple disconnected subtours (Figure 4c, subtours forming a single cycle) and complex disconnect subtours (Figure 4d, subtours formed by multiple connected cycles). Note that only disconnected subtours should be prevented as alpha or simple connected subtours can be part of feasible integer solutions. Next we discuss several strategies to define constraints (23) giving origin to the different PO-MP models, starting with single lot formulations and later advancing to multiple lot versions.

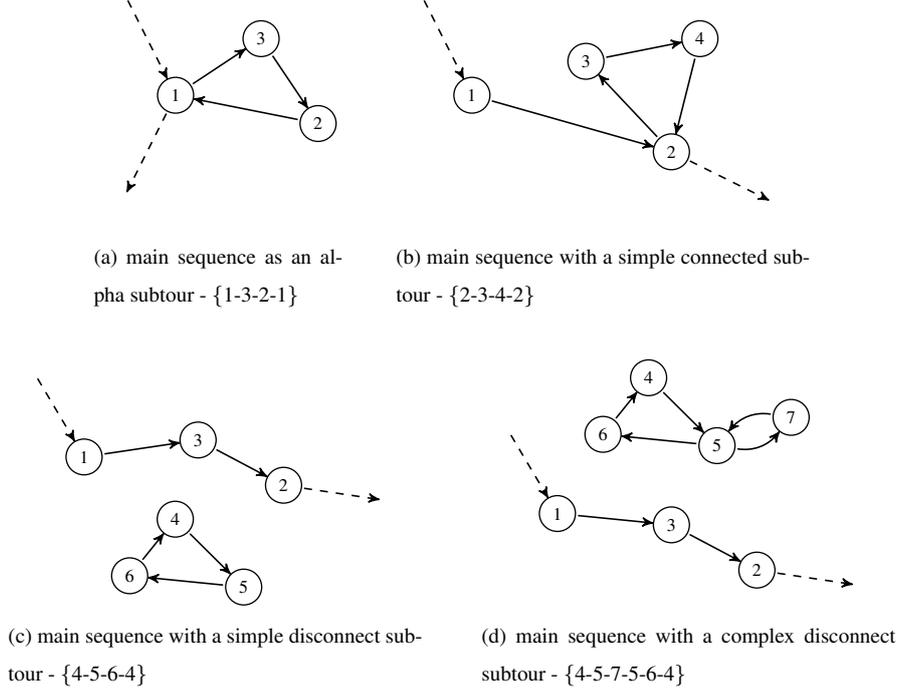


Figure 4: Examples of possible subtours.

4.2.1 Single lot

In the scope of single lot formulations by definition $q_{it} = 1$ for every i, t and, therefore, T_{ijt} are defined as binary, and connected and complex disconnect subtours are automatically excluded. We start by reviewing models which adopt constraints similar to those of the Miller, Tucker and Zemlin (MTZ) formulation of the Asymmetric Traveling Salesman Problem (ATSP). The first is the Smith-Daniels and Ritzman (SDR) formulation (Smith-Daniels and Ritzman (1988)), that uses C_{it} to map the completion time of the production lot of product i in period t . The subtour elimination constraints which replace (23) are stated as:

$$C_{jt} \geq C_{it} + st_{ij} + p_{jt} \cdot X_{jt} - M \cdot (1 - T_{ijt}) \quad \forall i, j \neq i, t, \quad (24)$$

$$C_{it} \geq p_{it} \cdot X_{it} \quad \forall i, t. \quad (25)$$

Constraints (24) guarantee that if product j follows product i in the sequence its completion time is greater than the completion time of product i plus the time required for setting up the machine and production. The completion time of the first production lot of each time period has to be also imposed, as done by (25). The authors did not present the above constraints as subtour elimination constraints, nevertheless they do translate into an accurate formulation of the problem. The SDR formulation can be further tighten by defining an upper bound of $C_{it} \leq cap_t$ to all completion times and by setting $M = st_{ij} + cap_t$. One of the advantages of computing the completion times comes from the possibility of imposing time windows to production lots or to synchronize parallel resources in multi-machine settings or multi-stage processes.

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

subtours. Constraints (23) are defined as:

$$V_{jt} \geq V_{it} + 1 - N \cdot (1 - T_{ijt}) \quad \forall i, j \neq i, t. \quad (26)$$

The above MTZ-based constraints (24) and (26) define sequences as paths. Therefore if a given product i is carried over from period $t - 1$, i.e. $Z_{it} = 1$, no setup can be performed to it during time period t . This follows from the fact that both sets of constraints imply $\sum_j T_{jit} + Z_{it} \leq 1$ and thus eliminate alpha subtours. Later Almada-Lobo et al. (2007) extended (26) in order to include alpha subtours. The reformulated constraints (26) present in Almada-Lobo et. al (AL1) are as follows:

$$V_{jt} \geq V_{it} + 1 - N \cdot (1 - T_{ijt}) - N \cdot Z_{it} \quad \forall i, j \neq i, t. \quad (27)$$

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

$$\sum_j T_{ijt} \leq 1 \quad \forall i, t, \quad (28)$$

$$\sum_j T_{jit} \leq 1 \quad \forall i, t. \quad (29)$$

Then constraints (27) are replaced by:

$$\sum_{i \in S, j \notin S} T_{ijt} + \sum_{i \in S} Z_{i,t+1} \geq \sum_j T_{jkt} \quad \forall t, k \in S, S \subseteq \mathcal{N}, \quad (30)$$

where \mathcal{N} is the power set of all products. In case of a potential simple disconnected subtour S the left-hand side of constraints (30) equals 0 while the right-hand side is at least 1, thus violating the inequality. Note that as shown in the cited work these constraints are still valid for alpha subtours.

4.2.2 Multiple lots

Considering multiple production lots of the same product within each time period is a non-trivial extension. To start, one has to deal with integer T_{ijt} variables, and, moreover, connected subtours are now allowed in sequences. The first formulation comes from Belvaux and Wolsey (2001) and resembles the prize collecting ATSP. Let $Y_{it} \in \{0, \dots, q_{it}\}$ be the number of setups to product i in period t . The Belvaux and Wolsey (BW) formulation is obtained by introducing new setup balance constraints (31)-(32) which replace (21) and the disconnect subtours elimination constraints listed as:

$$Z_{it} + \sum_j T_{jit} = Y_{it} \quad \forall i, t, \quad (31)$$

$$\sum_j T_{ijt} + Z_{i,t+1} = Y_{it} \quad \forall i, t, \quad (32)$$

$$\sum_{i,j \in S} T_{ijt} \leq \sum_{i \in S} Y_{it} - \frac{1}{q_{kt}} \cdot Y_{kt} \quad \forall t, k \in S, S \subseteq \mathcal{N}. \quad (33)$$

Subtour elimination constraints (33) build on the idea from the Dantzig, Fulkerson and Johnson (DFJ) formulation of the ATSP. When product k from the subset selected is in the sequence, these constraints establish that

the number of changeovers within the subset has to be less than the cardinality of the subset. The only exception occurs when one of the products is the first product in the sequence, in this case the number of setups within the group may equal the subset's cardinality. The complete formulation for BW reads (16) - (20), (31) - (33) and the updated variable domain.

The formulation by Menezes et al. (2011) (MCAL) also has an exponential number of inequalities to eliminate disconnected subtours. Let us define G_{it} to be a binary variable which takes the value one if the machine is set up for product i at least once in period t . The following constraints replace (23) to correctly model the problem:

$$\sum_j T_{jit} + Z_{it} \geq G_{it} \quad \forall i, t, \quad (34)$$

$$\sum_j T_{jit} + Z_{it} \leq q_{it} \cdot G_{it} \quad \forall i, t, \quad (35)$$

$$\sum_{i \in S, j \notin S} T_{jit} + \sum_{i \in S} Z_{it} \geq 1 - M \cdot \sum_{i \in S} (1 - G_{it}) \quad \forall t, k \in S, S \subseteq \mathcal{N}, |S| \geq 2. \quad (36)$$

Both (34) and (35) establish the correct relationship between the product setup state, changeovers and the initial machine configuration. In order to be active, inequalities (36) require all the products in subset S to be produced in time period t . When active, it imposes that the number of changeovers coming from products not belonging to the subset and/or the machine initial setup configuration to products within the subset has to be greater than one, thus connecting the subset. To reduce the number of inequalities required the authors also introduced *a priori* set of constraints that prevent simple disconnected subtours. For that propose let us define binary decision variable Q_{ijt} which equals 1 if at least one changeover from product i to product j is performed in period t . The additional constraints are as follows:

$$T_{ijt} \geq Q_{ijt} \quad \forall i, j, t, \quad (37)$$

$$T_{ijt} \leq q_{jt} \cdot Q_{ijt} \quad \forall i, j, t, \quad (38)$$

$$V_{jt} \geq V_{it} + 1 - M \cdot (1 - Q_{ijt}) - M \cdot \left(\sum_k T_{kit} + Z_{it} - Q_{ijt} \right) \quad \forall i, j \neq i, t. \quad (39)$$

The complete MCAL model is (16) - (22), (34) - (39) and the variable domain definition.

Commodity flow based formulations

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

The SCF model below is a new contribution of this work. The continuous variables F_{ijt} represent the flow of the commodity from node i to node j in period t . An artificial node indexed by 0 is introduced to capture the setup carryover acting as the source of the flow. Disconnected subtours are eliminated by ensuring the connectivity of

the graph induced by non-zero T 's. For this purpose, the following constraints enforce the existence of a path from the source to each product in the sequence:

$$\sum_j F_{0jt} = \sum_j G_{jt} \quad \forall t, \quad (40)$$

$$\sum_{j=0}^N F_{jit} = G_{it} + \sum_j F_{ijt} \quad \forall i, t. \quad (41)$$

$$F_{0it} \leq N \cdot Z_{it} \quad \forall i, t, \quad (42)$$

$$F_{ijt} \leq N \cdot T_{ijt} \quad \forall i, j, t. \quad (43)$$

Constraints (40) force the commodity flow to leave the source. The total flow amount is required to be equal to the number of products being produced in the period which is equivalent to the number of paths needed. The flow balance constraints are expressed by (41) which ensure that a unitary flow is sent to every selected node, corresponding to a path from the source to every product being produced in the time period. Both (42) and (43) impose an upper bound on the amount of flow traversing the arcs. Constraints (42) impose that the flow can only leave the source to the first product in the sequence, while (43) guarantee that the flow only traverses arcs in the current solution.

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

$$\sum_j F_{0jt}^k = G_{kt} \quad \forall k, t, \quad (44)$$

$$\sum_{j=0, j \neq k}^N F_{jit}^k = \sum_j F_{ijt}^k \quad \forall k, i \neq k, t \quad (45)$$

$$\sum_{j=0}^N F_{jkt}^k = G_{kt} \quad \forall k, t \quad (46)$$

$$F_{0jt}^k \leq Z_{jt} \quad \forall k, j, t, \quad (47)$$

$$F_{ijt}^k \leq T_{ijt} \quad \forall k, i, j, t. \quad (48)$$

Constraints (44) force the flow of commodity k to leave the source only if product k is part of the sequence. The flow conservation of commodity k is preserved by (45), while (46) force the path to end in product k . The flow can only use arcs in the current solution as imposed by (47) and (48).

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

5. Sequence oriented formulations

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

\widehat{sc}_s setup cost incurred if sequence s is selected,

\widehat{st}_s setup time incurred if sequence s is selected,

g_{is} =1 if product i is present in sequence s ,

f_{is} =1 if product i is first in sequence s ,

l_{is} =1 if product i is last in sequence s .

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

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

5.1 Micro-period models

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

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

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

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

$$\text{KMT} \quad \min \sum_{i,t} h_{it} \cdot I_{it} + \sum_s \widehat{sc}_s \cdot W_s \quad (49)$$

$$\text{s.t.} \quad I_{i,t-1} + X_{it} = d_{it} + I_{it} \quad \forall i, t, \quad (50)$$

$$\sum_i p_{it} \cdot X_{it} + \sum_{r \in L_t, s \in R_r} \widehat{st}_s \cdot W_s \leq cap_t \quad \forall t, \quad (51)$$

$$\sum_{s \in R_1} W_s = 1, \quad (52)$$

$$\sum_{s \in R_r} f_{is} \cdot W_s - \sum_{s \in R_{r-1}} l_{is} \cdot W_s = E_{i,r-1} - E_{ir} \quad \forall i, r \geq 2, \quad (53)$$

$$\sum_{s \in R_r} g_{is} \cdot W_s = Y_{ir} \quad \forall i, r, \quad (54)$$

$$X_{it} \leq b_{it} \cdot \sum_{r \in L_t} Y_{ir} \quad \forall i, t, \quad (55)$$

$$X_{it} \geq m_i \cdot \sum_{r \in L_t} Y_{ir} \quad \forall i, t, \quad (56)$$

$$X, I, W, E \geq 0, \quad Y \in \{0, 1\}. \quad (57)$$

Objective function (49) minimizes the total expenditure in holding costs and setup costs incurred from sequence selection. Constraints (50) represent the classical inventory balance constraints. Note that production is not divided among the different split-sequences as one may expect in mP models. Capacity constraints are expressed by (51) in which the total setup time within the period is obtained by summing the setup times incurred in each split-sequence. Sequence selection in split-sequences is ruled by (52) and (53). The first constraint imposes that a sequence must be chosen for the first split-sequence. Constraints (53) link sequence selection among adjacent split-sequences, moreover, they also preserve setup state when empty split-sequences occur. The relationship between product setup state in each split-sequence and sequence selection is guaranteed by (54). Finally, (55) and (56) define the bounds on production in each period according to the product setup state in the corresponding split-sequences.

5.2 Macro-period models

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

5.2.1 Single lot

The model proposed in Haase and Kimms (2000) (HK) is obtained by adding to the constraints below the requirements (49) and (50).

$$\sum_i p_{it} \cdot X_{it} + \sum_{s \in S_t} \hat{st}_s \cdot W_s \leq cap_t \quad \forall t, \quad (58)$$

$$\sum_{s \in S_t} W_s = 1 \quad \forall t, \quad (59)$$

$$\sum_{s \in S_t} f_{is} \cdot W_s = \sum_{s \in S_{t-1}} l_{is} \cdot W_s \quad \forall i, t, \quad (60)$$

$$X_{it} \leq b_{it} \cdot \sum_{s \in S_t} g_{is} \cdot W_s \quad \forall i, t, \quad (61)$$

$$X, I \geq 0, \quad W \in \{0, 1\}. \quad (62)$$

The new capacity constraints are expressed in (58). The use of a single sequence in each period is ensured by (59), while (60) guarantee setup carry-over by linking the first and last products of consecutive time periods. The last set of constraints (61) only allows production for products in the sequence selected. Similarly to the original

paper, the model HK is consider a single lot model. Nevertheless, it is important to note that by changing the definition of g_{is} to the number of times product i appears in sequence s , the model tackles multiple lots of the same product.

5.2.2 Multiple lots

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

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

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

We also need to introduce binary decision variables U_{it} to capture if at least one setup is performed to product i in period t . The overall GKAL models contains (49), (50), (58) together with:

$$\sum_{s \in \mathcal{S}_t} f_{is} \cdot W_s = Z_{it} \quad \forall i, t, \quad (63)$$

$$\sum_{s \in \mathcal{S}_t} l_{is} \cdot W_s = Z_{i,t+1} \quad \forall i, t, \quad (64)$$

$$\sum_i Z_{it} = 1 \quad \forall t, \quad (65)$$

$$\sum_{s \in \mathcal{S}_t} e_{is} \cdot W_s = U_{it} \quad \forall i, t, \quad (66)$$

$$\sum_{s \in \mathcal{S}_t} a_{is} \cdot W_s = Y_{it} \quad \forall i, t, \quad (67)$$

$$X_{it} \leq b_{it} \cdot (U_{it} + Z_{it}) \quad \forall i, t, \quad (68)$$

$$X, I, W \geq 0, \quad Z, U \in \{0, 1\}, \quad Y_{it} \in \{0, \dots, q_{it}\}. \quad (69)$$

The first two set of constraints (63) and (64) link the machine's initial configuration in each period with the first and last product in the selected sequence, and also establish the setup carry-over. Constraints (65) state that the machine is set up for exactly one product at the beginning of each time period. Product setup decisions are linked with sequence selection through constraints (66) and (67). Requirements (68) ensure for each period that a product is only produced in the case the machine is properly set-up. Such a configuration might have been carried over from the previous period or resulted from a setup in that period.

6. Computational tests results

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

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

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

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

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

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

In the following subsections results are divided according to the time structure of the models. In the first benchmark we explore the effect of the different formulations in mP models. We do not compare micro and

macro-period models since their comparison from previous studies (Menezes et al. (2011)) has already been established and it can also be drawn from the results of the following subsection that mP models struggle even when the instances are of medium size. The second benchmark is fully dedicated to MP models. We start by assessing the models on a set of instances obeying the triangle inequality and later the formulations are tested on a non-triangular instance set obtained by modifying these instances and adding minimum lot size requirements.

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

6.1 Micro-period models

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

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

The comparison of the several mP models is shown in Figure 5. On the horizontal axis we have the mean running time in seconds and on the vertical axis the mean deviation from the best known solution, which are actual optimal values available from Menezes et al. (2011). Hence, the closer from the bottom left corner of the chart the more effective/efficient the model is. The original formulations are depicted as squares and the FL reformulations as circles.

Let us first discuss the results obtained by GLSP. Both the original and reformulated versions often exceed the available memory during the tree search performed by CPLEX, which is reflected in the low running times. The memory limit is exceeded in 3 (out of 12) instances by the original version and in 10 instances by the reformulation. In line with this fact, the GLSP has the worst performance among the mP models in terms of final solution quality

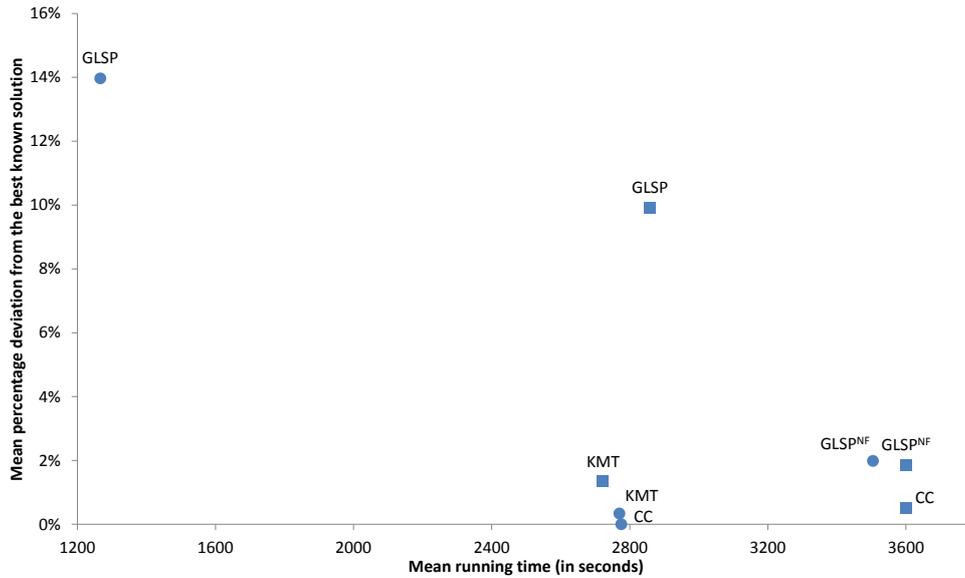


Figure 5: Performance comparison of mP formulations

as the tree search is often prematurely stopped. Considering PO-mP original formulations, the CC model has the best performance and the network reformulation of the GLSP clearly improves its solution quality, besides improving memory consumption. Note that the original versions of GLSP^{NF} and CC consume all the available running time in the tree search, thus not being able to prove optimality in any of the instances. In terms of original formulations, the SO model KMT has the best performance among all delivering superior results in terms of solution quality and running times.

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

6.2 Macro-period models

6.2.1 Triangular instances

The MP models are first tested on a problem set obeying the triangle inequality and available from James and Almada-Lobo (2011) which adapts the approach of Almada-Lobo et al. (2007) to consider different values for capacity utilization over the planning horizon. All data parameters are generated from a uniform distribution.

Product demand ranges between 40 and 59 units per period, holding costs between 2 and 9 cost units per period and setup times vary between 5 and 10 time units. Setup costs are made proportional to setup times by using a cost factor θ . The processing time is equal to all products and set to one time unit. To define machine capacity two parameters Cut and $CutVar$ are used. Cut establishes the target machine utilization over the entire planning horizon and $CutVar$ controls the maximum deviation from the target capacity utilization in each period. Moreover, it is ensured that the cumulative capacity utilization in any period does not exceed Cut in order to ensure problem feasibility.

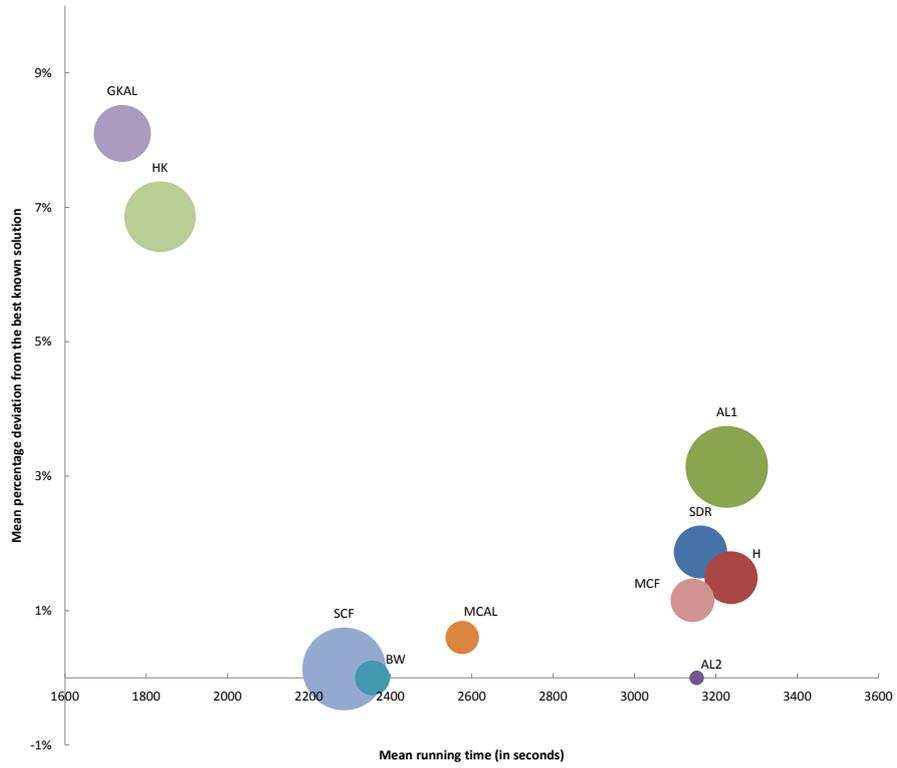
Instances are classified into problem types according to the five-tuple $(N, T, Cut, CutVar, \theta)$. We use a total of 160 instances, 10 different instances for each one of the 16 problem types created by combining the following values for the parameters: $N \in \{15, 25\}$, $T \in \{10, 15\}$, $Cut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

Figure 6 presents the comparison of several MP models on the triangular instance set separated into the original formulation and the reformulation. The size of each circle accounts for the number of instances in which the model is able to provide at least one feasible solution. Hence, the larger the circle the more problems are solved. The horizontal and vertical axis measure the mean running time in seconds and the mean deviation from the best known solution, respectively. The mean deviation only takes into consideration deviations in problems with feasible solutions. The best known solution is the best objective function among the solutions of all the models, including the reformulated versions, and is often the provably optimal solution to the problem (81 out of 160 instances).

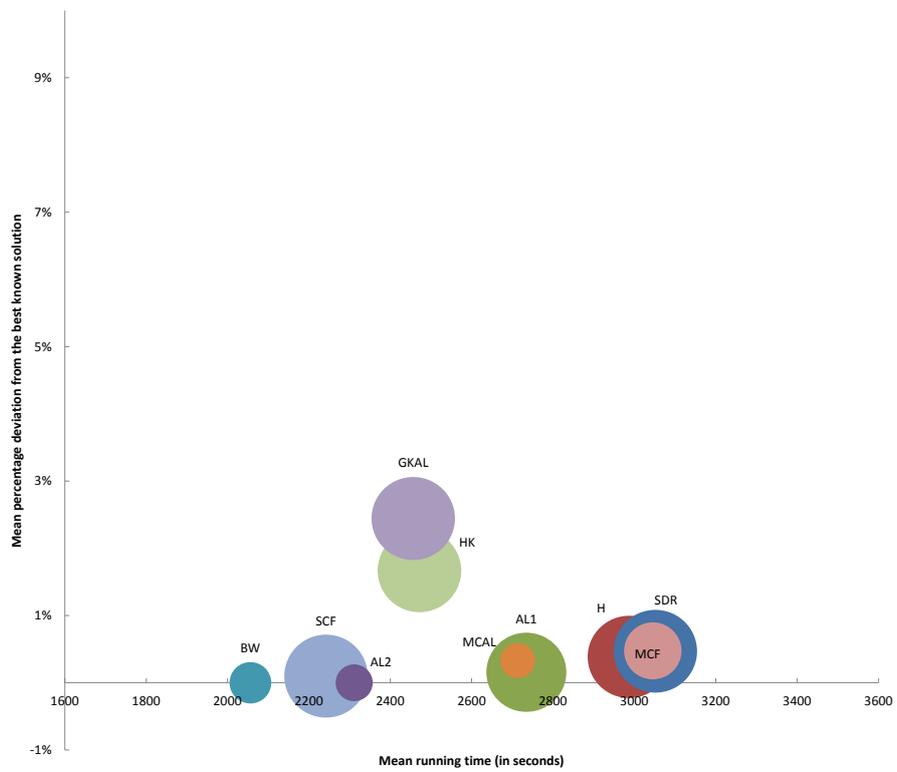
Concerning the original formulations, it is clear that PO-MP-SL models have a poor performance in terms of the running time. We highlight three results: (1) SDR, H and AL1 exceed the memory limit in 39, 32 and 28 instances, respectively, explaining the mean running time below one hour, since they rarely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

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

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



(a) Original formulation



(b) FL reformulation

Figure 6: Performance comparison of MP formulations on the triangular instance set

time is used for branching to prove solution optimality or to find a very good integer feasible solution.

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

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

Overall, both versions of SCF exhibit the overall best trade-off providing superior quality feasible solutions to all the problems in reasonable running times. Moreover, for the two versions of SCF the final MIP gap is on average less than %0.1 emphasizing the quality of the solutions provided. We also draw attention to BW which finds the largest number of optimal solutions, a total of 80 in less than 520 seconds with the FL reformulation. The full results on this instance set is shown in Table C.3 in Appendix Appendix C.

6.2.2 *Non-triangular instances*

To test the MP models on large instances disobeying the triangle inequality, we use the problem set designed by Guimarães et al. (2013) that is based on the set of triangular instances described in the previous section. To induce non-triangular setup matrices in the original problem set, the authors modify setup times of a subset of products called hereafter shortcut products. For each one of the k shortcut products new setup times st_{ik} and st_{ki} were randomly generated from a uniform distribution between 2 and 4, while setup costs remain proportional to setup times using the cost factor θ . A total of 2 and 3 shortcut products are introduced in instances with 15 and 25

products, respectively. The problem set is composed by the same 160 problems grouped in 16 problem types, but with the setup matrices modified. A minimum lot size of 25 units was also introduced to avoid having fictitious setups at optimal solutions.

Figure 7 shows the comparison of several MP models for the non-triangular instance set in the same format as the comparison made for the triangular set. In this set a total of 98 solutions were proved to be optimal.

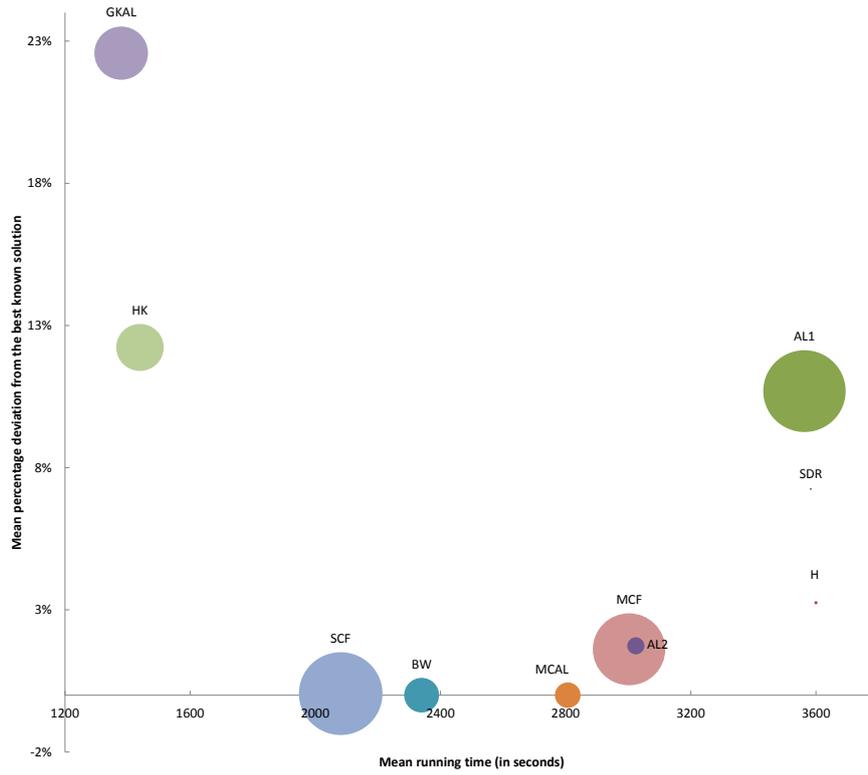
Models SDR and H and their respective reformulations fail to identify a feasible solution in more than 90% of the problems, and the tractability of the reformulated models is an important issue since the memory limit is reached in 79 cases for the SDR and 72 for the H model. In this problem set extra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

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

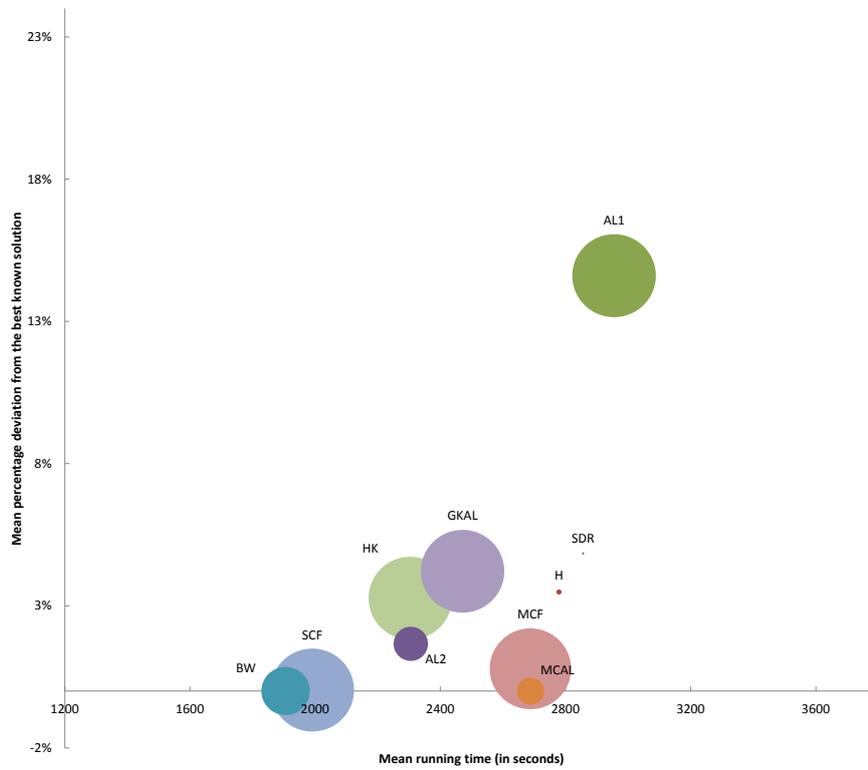
Mean deviation from the best known solution of SO-MP models considerably worsens, especially in the original formulation suggesting a lower quality of the relaxation in the presence of non-triangular setups and minimum lotsizes. Applying the FL reformulation to these models results in the same performance change as in the triangular set, increasing the running times and feasible solutions and decreasing the mean deviation. Again, GKAL shows a bigger improvement, but is still not enough to match HK's performance. Table C.4 in Appendix Appendix C details these results.

7. Conclusions

In this paper a two-dimensional framework is proposed to review and classify the different modeling approaches to incorporate sequencing decision in lotsizing and scheduling models. The framework uses the sequencing technique and time structure dimensions to divide the approaches into classes. The most relevant models in each class are reviewed to present their main features and differences, especially in the underlying assumptions.



(a) Original formulation



(b) FL reformulation

Figure 7: Performance comparison of MP formulations on the non-triangular instance set

From this study emerged an important contribution which is a new polynomially sized formulation to the problem using commodity flow based subtour elimination constraints.

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

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

Acknowledgments

The first author is grateful to the Portuguese Foundation for Science and Technology for awarding him a grant (SFRH/BD/62010/2009).

Appendix A. Relationships among product oriented small bucket models

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

Proof. Let (X, I, T) be an optimal solution to the LP relaxation of CC. We define $Y_{jn} = \sum_i T_{ijn}$ for every j, n and show that (X, I, T, Y) is a feasible solution to the LP relaxation of (1) - (8) with the same objective value. Constraints (2), (3), (4) hold by definition.

Summing (13) over all i we obtain

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

This together with (11) implies

$$\sum_{i,j} T_{jin} = 1 \quad \forall n \quad (\text{A.1})$$

which is equivalent to $\sum_j Y_{jn} = 1$ and thus (5).

To show (7), observe that (13) implies

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

Now we get

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

which immediately implies (7).

Finally, note that (6) is equivalent to

$$1 \geq Y_{i,n-1} + Y_{in} - T_{ijn} = \sum_k T_{ki,n-1} + \sum_t T_{tjn} - T_{ijn} = \quad (\text{A.2})$$

$$\sum_s T_{isn} + \sum_t T_{tjn} - T_{ijn} = \sum_s T_{isn} + \sum_{t,t \neq i} T_{tjn} \quad (\text{A.3})$$

where (A.2) follow by (13).

Constraint (A.1) implies

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

which is identical to (A.3). This in turn shows (6). \square

Appendix B. Subproblem formulation

The subproblem arising in each time period in models KMT, HK and GKAL resembles the prize collecting traveling salesman problem introduced by Balas (1989). Network $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ consists of node set $\mathcal{V} = \mathcal{N} \cup \{0, N+1\}$ and arc set \mathcal{A} . Node 0 is the source and node $N+1$ the sink while the remaining nodes represent products (see Figure B.8). The source and the sink are used to identify the starting and ending products of the sequence, hence an arc connecting the source to a product means a carry over from the previous period and, similarly, an arc connecting a product to the sink represents a carry over to the next period. Travel costs c_{ij} are incurred for traversing arcs (i, j) and a prize ρ_i for including node i in the walk. The objective is to find the minimum cost walk through the network from the source to the sink.

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

$$(\text{sub}_t) \quad \min \sum_{i,j \in \mathcal{V}} c_{ij} \cdot \chi_{ij} - \sum_{i \in \mathcal{N}} \rho_i \cdot y'_i \quad (\text{B.1})$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{V}} \chi_{ji} = \sum_{j \in \mathcal{V}} \chi_{ij} \quad \forall i \in \mathcal{N}, \quad (\text{B.2})$$

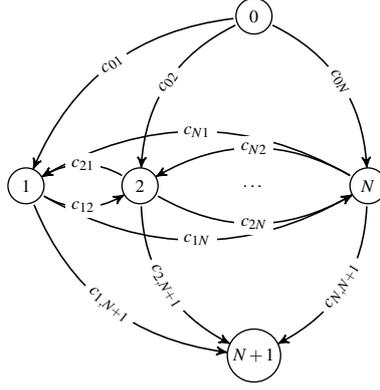


Figure B.8: Network representation of the subproblem

$$\sum_{j \in \mathcal{N}} \chi_{0j} = 1, \quad (\text{B.3})$$

$$\sum_{j \in \mathcal{N}} \chi_{j,N+1} = 1, \quad (\text{B.4})$$

$$y'_i \leq \sum_{j \in \mathcal{V}} \chi_{ji} \quad \forall i \in \mathcal{N}, \quad (\text{B.5})$$

$$y'_i \geq \sum_{j \in \mathcal{V}} \chi_{ji} \quad \forall i \in \mathcal{N}, \quad (\text{B.6})$$

$$\chi_{ij} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A}. \quad (\text{B.7})$$

Here c_{ij} and ρ_i are derived from pricing equation. Here c_{ij} and ρ_i are derived from the pricing equation not listed herein. Objective function (B.1) minimizes the cost of the traversed arc minus the prizes collected from the scheduled products (visited nodes). Constraints (B.2) balance in- and out-flow of each product. The source and sink nodes must be connected to a product, guaranteed by requirements (B.3) and (B.4), representing the first and last products in the sequence. The last two sets of constraints (B.5)-(B.6) represent the logical connections between node variables.

The model for the subproblems is, however, still incomplete, as a solution for (B.1)-(B.7) permits disconnected subtours. To eliminate such subtours we use single-commodity-flow type constraints. Consider decision variables f_{ij} as the commodity flow traversing arc (i, j) , which is constrained to be less than or equal to the number of products. The following constraints are added to sub_t to prohibit disconnected subtours:

$$f_{ij} \leq N \cdot \chi_{ij} \quad \forall i \in \mathcal{V}, j \in \mathcal{N}, \quad (\text{B.8})$$

$$\sum_{j \in \mathcal{N}} f_{0j} = \sum_{j \in \mathcal{N}} y'_j, \quad (\text{B.9})$$

$$\sum_{j \in \mathcal{V}} f_{ji} = y'_i + \sum_{j \in \mathcal{N}} f_{ji} \quad \forall i \in \mathcal{N}. \quad (\text{B.10})$$

Constraints (B.8) ensure that flows only traverse the arcs in the solution. Constraints (B.9)-(B.10) require that flow variables of the commodity describe a path from the source to every node in the sequence defined by arc variables. In detail, constraints (B.9) force a flow equal to the number of products in the sequence to leave the source and constraints (B.10) impose flow conservation for each node in the graph.

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

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

$$(sub_t^{KMT}) \quad \min \quad \sum_{i,j \in \mathcal{V}} c_{ij} \cdot \chi_{ij} + \sum_{i,j \in \mathcal{N}} c'_{ij} \cdot \chi'_{ij} \quad (\text{B.11})$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{V}} \chi_{ji} + \sum_{j \in \mathcal{N}} \chi'_{ji} = \sum_{j \in \mathcal{V}} \chi_{ij} + \sum_{j \in \mathcal{N}} \chi'_{ij} \quad \forall i \in \mathcal{N}, \quad (\text{B.12})$$

$$\sum_{j \in \mathcal{N}} \chi_{ij} \leq 1 \quad \forall i \in \mathcal{N} \quad (\text{B.13})$$

$$\sum_{j \in \mathcal{V}} \chi_{ji} \leq 1 \quad \forall i \in \mathcal{N} \quad (\text{B.14})$$

$$\sum_{i,j \in \mathcal{N}} \chi_{ij} \leq Br^{max} - 1 \quad \forall i, j \in \mathcal{N}, \quad (\text{B.15})$$

$$\sum_{j \in \mathcal{N}} \chi'_{ji} \leq \chi_{i,N+1} \quad \forall i \in \mathcal{N}, \quad (\text{B.16})$$

$$\sum_{i,j \in \mathcal{N}} \chi'_{ij} = 1, \quad (\text{B.17})$$

$$(\text{B.3}) - (\text{B.6}), (\text{B.8}) - (\text{B.10}),$$

$$\chi_{ij} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A}, \quad (\text{B.18})$$

$$\chi'_{ij} \in \{0, 1\} \quad \forall i, j \in \mathcal{N}. \quad (\text{B.19})$$

The last changeover variables are introduced into objective function (B.11) and similarly to HK, no prizes appear in the pricing equations. Setup conservation constraints (B.12) now include the last changeover. The maximum number of Br^{max} in the sequence is guaranteed by allowing up to $Br^{max} - 1$ setups to take place through constraints (B.15). The correct linking between the last setup and final machine configuration is ensured by (B.16) and constraints (B.17) impose the last changeover to take place.

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

$$(sub_t^{GKAL}) \quad \min \quad \sum_{i,j \in \mathcal{V}} c_{ij} \cdot \chi_{ij} + \sum_{i \in \mathcal{N}} \rho_i \cdot y_i \quad (\text{B.20})$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{N}} \chi_{ji} \geq y_i \quad \forall i \in \mathcal{N}, \quad (\text{B.21})$$

$$\sum_{j \in \mathcal{N}} \chi_{ji} \leq q_{it} \cdot y_i \quad \forall i \in \mathcal{N}, \quad (\text{B.22})$$

$$y'_i \leq y_i + \chi_{0i} \quad \forall i \in \mathcal{N}, \quad (\text{B.23})$$

$$2 \cdot y'_i \geq y_i + \chi_{0i} \quad \forall i \in \mathcal{N}, \quad (\text{B.24})$$

(B.3) – (B.4), (B.8) – (B.10),

$$\chi_{ij} \in \mathbb{N} \quad \forall (i, j) \in \mathcal{A}, \quad (\text{B.25})$$

$$y_i \in \{0, 1\} \quad \forall i \in \mathcal{N}. \quad (\text{B.26})$$

Objective function (B.20) minimizes the cost of the traversed arc minus the prizes collected from the scheduled products (visited nodes). Constraints (B.21) and (B.22) enforce the logical relationship between the arcs traversed and nodes visited. The difference between y_i and y'_i relies on the fact that the latter equals to one also if the product is scheduled immediately after the source (first in the sequence, not representing an actual setup into it). Constraints (B.23)-(B.24) represent the logical connections between node variables y and y' .

Appendix C. Results Tables

Table C.2: Summary of results for the mP models. The first row in each model corresponds to the original formulation and the second to the FL reformulation.

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

Table C.3: Summary of results for MP models for the triangular instance set. The first row in each model corresponds to the original formulation and the second to the FL reformulation.

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

Table C.4: Summary of results for MP models for the non-triangular instance set. The first row in each model corresponds to the original formulation and the second to the FL reformulation.

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

References

- B. Almada-Lobo, D. Klabjan, J. F. Oliveira, and M. A. Carravilla. Single machine multi-product capacitated lot sizing with sequence-dependent setups. *International Journal of Production Research*, 45(20):4873 – 4894, 2007.
- B. Almada-Lobo, D. Klabjan, M. Carravilla, and J. Oliveira. Multiple machine continuous setup lotsizing with sequence-dependent setups. *Computational Optimization and Applications*, 47(3):529–552, 2010.
- E. Balas. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636, 1989.
- G. Belvaux and L. A. Wolsey. Modelling practical lot-sizing problems as mixed-integer programs. *Management Science*, 47(7):993–1007, 2001.
- A. Clark, B. Almada-Lobo, and C. Almeder. Lot sizing and scheduling: industrial extensions and research opportunities. *International Journal of Production Research*, 49(9):2457–2461, 2011.
- A. R. Clark and S. J. Clark. Rolling-horizon lot-sizing when set-up times are sequence-dependent. *International Journal of Production Research*, 38(10):2287–2307, 2000.
- A. Drexl and K. Haase. Proportional lotsizing and scheduling. *International Journal of Production Economics*, 40(1):73 – 87, 1995.
- A. Drexl and A. Kimms. Lot sizing and scheduling - survey and extensions. *European Journal of Operational Research*, 99(2):221–235, 1997.
- B. Fleischmann. The discrete lot-sizing and scheduling problem with sequence-dependent setup costs. *European Journal of Operational Research*, 75(2):395 – 404, 1994.
- B. Fleischmann and H. Meyr. The general lotsizing and scheduling problem. *OR Spektrum*, 19(1):11–21, 1997.
- L. Guimarães, D. Klabjan, and B. Almada-Lobo. Pricing, relaxing and fixing under lot sizing and scheduling. *European Journal of Operational Research*, (0):–, 2013. doi: 10.1016/j.ejor.2013.04.030. Available in Press.
- K. Haase. Capacitated lot-sizing with sequence dependent setup costs. *OR Spectrum*, 18(1):51–59, 1996.
- K. Haase and A. Kimms. Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. *International Journal of Production Economics*, 66(2):159 – 169, 2000.
- R. J. James and B. Almada-Lobo. Single and parallel machine capacitated lotsizing and scheduling: New iterative MIP-based neighborhood search heuristics. *Computers & Operations Research*, 38(12):1816 – 1825, 2011.
- R. Jans and Z. Degraeve. Modeling industrial lot sizing problems: a review. *International Journal of Production Research*, 46(6):1619–1643, 2008.
- S. Kang, K. Malik, and L. J. Thomas. Lotsizing and scheduling on parallel machines with sequence-dependent setup costs. *Management Science*, 45(2):273–289, 1999.
- J. Krarup and O. Bilde. Plant location, set covering and economic lot size: An O(mn) algorithm for structured problems. In *Optimierung bei Graphentheoretischen und Ganzzahligen Probleme*, pages 155 – 180. BirkhauserVerlag, Berlin, 1977.
- A. Menezes, A. Clark, and B. Almada-Lobo. Capacitated lot-sizing and scheduling with sequence-dependent, period-overlapping and non-triangular setups. *Journal of Scheduling*, 14(2):209–219, 2011.
- H. Meyr. Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. *European Journal of Operational Research*, 120(2):311–326, 2000.
- H. Meyr. Simultaneous lotsizing and scheduling on parallel machines. *European Journal of Operational Research*, 139(2):277 – 292, 2002.
- T. Öncan, I. K. Altinel, and G. Laporte. A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3):637 – 654, 2009.
- D. Quadt and H. Kuhn. Capacitated lot-sizing with extensions: a review. *4OR: A Quarterly Journal of Operations Research*, 6(1):61–83, 2008.
- S. Sarin, H. Sherali, and L. Yao. New formulation for the high multiplicity asymmetric traveling salesman problem with application to the Chesapeake problem. *Optimization Letters*, 5(2):259–272, 2011.
- V. L. Smith-Daniels and L. P. Ritzman. A model for lot sizing and sequencing in process industries. *International Journal of Production Research*, 26(4):647–674, 1988.
- L. A. Wolsey. MIP modelling of changeovers in production planning and scheduling problems. *European Journal of Operational Research*, 99(1):154 – 165, 1997.
- X. Zhu and W. E. Wilhelm. Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE Transactions*, 38(11):987–1007, 2006.