

Capacitated Lotsizing and Scheduling with Sequence-dependent, Period Overlapping and Non Triangular Setups

António Aroso Menezes^{a,b}
Alistair Clark^b Bernardo Almada-Lobo^{a,*}

^a Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias s/n, Porto 4200-465, Portugal

^b Bristol Institute of Technology, University of the West of England, Coldharbour Lane, Bristol BS16 1QY, United Kingdom

May 3, 2010

Abstract

In production planning, sequence dependent setup times and costs are often incurred for switchovers from one product to another. When setup times and costs do not respect the triangular inequality, a situation may occur where the optimal solution includes more than one batch of the same product in a single period — in other words, at least one sub tour exists in the production sequence of that period. By allowing setup crossovers, flexibility is increased and better solutions can be found. In tight capacity conditions, or whenever setup times are significant, setup crossovers are needed to assure feasibility. We present the first linear mixed integer programming extension for the capacitated lotsizing and scheduling problem incorporating all the necessary features of sequence sub tours and setup crossovers. This formulation is more efficient than other well known lotsizing and scheduling models.

1 Introduction

In production planning problems, the timing and sizes of production orders of a given range of products over a certain number of periods must be determined so that market demand is fulfilled and associated costs are minimized. In a large number of cases, sequence dependent setup times and costs are incurred for switchovers from one product to another. In addition, tight machine capacity often necessitates product inventories from one period to another, further increasing costs.

Usually, setups follow the so called *triangular inequality*, *i.e.*, for any three products, the cost and time required to directly set up the machine from one product to another is always less than the sum of those required when setting up via an intermediate product. However, several reasons may lead to the existence of non-triangular setup times or costs. For example, in some industries (chemical, pharmaceutical, food, dyeing, etc.), unwanted

*Corresponding author. *E-mail address*: almada.loba@fe.up.pt; tel: +351225082133

contamination occurs between certain products. To avoid it, additional cleansing operations must be performed during machine set ups, often requiring the use of expensive special products. Alternatively, products that absorb the contaminating substances (or even lower grade mixed products) can be produced in between, reducing setup times and costs. In this situation, a minimum lot size is often required for the intermediate product, so as to eliminate unacceptable contamination.

When the triangular inequality does not hold, it may be efficient to produce more than one batch of the same product in a given period. Though this concept is quite simple, it is also quite hard to model when considering sequence dependent setup times and costs, often leading to incomplete or inaccurate MIP formulations, and subsequent infeasible or sub optimal solutions. In this paper, a novel formulation that correctly handles this problem is proposed.

Setup times play a double role in production planning problems, because not only do they indirectly impact on the optimal solution's value (by constraining the inventory levels), but they also interfere with the solution's feasibility. In tight machine capacity situations, an effective distribution of setup times is required. Quite often, setup operations can be interrupted at the end of a period and resumed at the beginning of the next one with no additional prejudice, either due to the nature of the operations required or to the fact that there is no physical separation between periods. When this happens, a *setup cross over* or a *period – overlapping setup* is said to exist.

To the best of our knowledge, we are the first to exactly model these features as an extension of the well known capacitated lotsizing problem (CLSP), considered to be a big-bucket model as multiple products/setup may be produced/performed per planning period. We show that such formulations are more efficient than standard small-bucket models.

In section 2 we introduce a previously existing model for lotsizing with sequence dependent setup costs and times. We then expand this model so as to correctly handle non-triangular setup costs and times, with the necessary feature of disconnected sub-tour elimination and minimum lot size enforcement. Furthermore, we benchmark our new formulation against the mixed small-bucket-big-bucket general lotsizing and scheduling model on reported instances from the literature. In section 3 we present a new formulation extension that allows setups to crossover between adjacent periods. Numerical examples and visual representations are provided in sections 2 and 3 to demonstrate all features. Finally, we summarize our main findings. We conclude this introduction with a literature review.

1.1 Literature Review

There is a vast amount of literature on lotsizing and scheduling models with setup times and costs. However, to the best of our knowledge, none is able to correctly handle both sequence dependent and non triangular setups for big-bucket models, let alone a combination of this and period overlapping setups.

Several contributions from the academic community have greatly extended and improved the quality of CLSP models. Gopalakrishnan et al. (1995) are among the first to address the influence of significant setup times, proposing a model that handles setup carry overs for identical times and costs. Later, in Gopalakrishnan (2000), product dependent setup times and costs are considered. Haase (1996) is the first to propose in the literature the big-bucket

capacitated dynamic lot-sizing problem in which setup costs are sequence dependent, but neglecting setup times (the so-called CLSD).

Sox and Gao (1999) propose a new model that only considers product dependent setup costs, but at the same time greatly reduces the number of binary variables, increasing the model's efficiency. Porkka and Kuula (2000) show that proper accounting for setup carryovers and setup times decrease the number of setups and frees a significant amount of production capacity. Suerie and Stadtler (2003) suggest a new model, considering both setup times and costs while keeping the number of binary variables low. Extensive computational tests prove the model's superiority with respect to previously existing models.

Haase and Kimms (2000) take a different approach, considering a CLSP-like model that handles sequence dependent setup times and costs, but by pre-defining efficient production sequences, sub-optimal solutions may be found. The authors also assume no inventory may exist at the beginning of the period in which a production lot of that same product is to be produced.

Clark and Clark (2000) model the CLSD with sequence-dependent setup times using a new mixed-integer programming formulation. They assume that up to a given number of setups occur in the time period between any two given products, independently of their demand patterns.

Almada-Lobo et al. (2007) propose two models that correctly handle sequence-dependent setup times and costs for large-bucket problems (several products/setups may be produced/performed per period), but do not allow setup cross overs, and may result in sub optimal solutions when non triangular setup times and costs exist.

CLSP partitions the planning horizon into a small number of lengthy time periods, allowing for the set up of several products within the same bucket. Small-bucket models divide the planning horizon into many short periods (such as days, shifts or hours), in which at most one setup may be performed. Such models may potentially take into account more "fine-grained" details.

In Toso and Morabito (2005), periods are divided in subperiods, enabling the original big-bucket problem to be treated as a small-bucket one, where at most one setup may be performed per subperiod. This way, sequence dependent setup times (either respecting or not the triangular inequality) are correctly accounted for. However, the subdivision requires an *a priori* definition of the maximum allowed number of setup operations per period, thus heavily increasing the model size, or even demanding multiple tweaking experiments before obtaining the optimal solution. While minimum lot sizes are imposed, they only relate to the first subperiod after the machine has been set up, leading to potential sub-optimal solutions. Moreover, setup cross overs are not allowed, having to start and finish in the same period.

Suerie (2006) proposes a model that correctly handles setup cross overs, but only for small-bucket problems. A set of variables are introduced in relation to the standard model, keeping track of how much time each setup operation is performed in each period, the cumulative time a setup operation has been performed in any given period since the last time it started, and the availability of the machine at the beginning of every period (the machine is available for production if the entire setup operation has finished). Sung and Maravelias (2008) propose a similar model for large-bucket problems, with sequence independent setup times and costs.

The reader is referred to Zhu and Wilhelm (2006) for an extensive literature review related to lotsizing and scheduling problems with sequence-dependent setups.

2 New Model for CLSP with Non Triangular Setup Costs and Times

2.1 Standard CLSD model with sequence dependent setup costs and times

Consider the following standard model for the CLSD with sequence-dependent setup costs and times, suggested by Almada-Lobo et al. (2007). This problem is essentially modeled in much the same way as the prize collecting salesman (see (Balas (1989)) or the vehicle routing problem (Laporte (1992b)) with additional sub tour elimination constraints. Here, t denotes time periods ranging from 1 to T , while i and j index the products, which are labeled from 1 to N . Furthermore, the set $\{1, 2, \dots, M\}$ is denoted by $[M]$. A general single-stage model is considered, involving multiple items to be scheduled on a single machine with the following data:

h_i	cost of carrying one unit of stock of product i from one period to the next,
p_i	processing time of one unit of product i ,
d_{it}	demand for product i at the end of period t ,
C_t	capacity of the machine in period t (measured in time units),
s_{ij}	time needed to set up the machine from product i to product j ,
c_{ij}	cost incurred to set up the machine from product i to product j ,
M_{it}	upper bound on the production quantity of product i in period t .

Binary variable T_{ijt} indicates whether or not a setup occurs on the machine configuration state from product i to j in period t . Continuous variable α_{it} keeps track of the machine state — if it is set up for product i (value 1) or not (value 0) — at the beginning of period t . Variable X_{it} represents the amount of product i to produce in period t , and I_{it} the stock of product i at the end of period t . Finally, auxiliary continuous variable V_{it} schedules production lot of product i in period t . The larger V_{it} , the later the product i is scheduled in period t , assuring that the machine is only set up for one product on any given time.

Lastly, v denotes optimal values of underlying optimization problems. This formulation, F_1 , will be used as a starting point for the extensions presented later in this paper:

$$v(F_1) = \min \sum_i \sum_j \sum_t c_{ij} \cdot T_{ijt} + \sum_i \sum_t h_i \cdot I_{it} \quad (1)$$

$$I_{it} = I_{i(t-1)} + X_{it} - d_{it} \quad i \in [N], t \in [T] \quad (2)$$

$$\sum_i p_i \cdot X_{it} + \sum_i \sum_j s_{ij} \cdot T_{ijt} \leq C_t \quad t \in [T] \quad (3)$$

$$X_{it} \leq M_{it} \cdot \left(\sum_j T_{jit} + \alpha_{it} \right) \quad i \in [N], t \in [T] \quad (4)$$

$$\sum_i \alpha_{it} = 1 \quad t \in [T] \quad (5)$$

$$\alpha_{it} + \sum_j T_{jit} = \alpha_{i(t+1)} + \sum_j T_{ijt} \quad i \in [N], t \in [T] \quad (6)$$

$$V_{it} + N \cdot T_{ijt} - (N - 1) - N \cdot \alpha_{it} \leq V_{jt} \quad \begin{array}{l} i \in [N], j \in [N] \setminus \{i\}, \\ t \in [T] \end{array} \quad (7)$$

$$(X_{it}, I_{it}, \alpha_{it}, V_{it}) \geq 0, T_{ijt} \in \{0, 1\}. \quad (8)$$

The objective function (1) minimizes the sum of sequence-dependent setup costs and the holding cost. Constraints (2) represent the inventory balances and (3) ensure that production and setup operations do not exceed available capacity. Constraints (4) guarantee that a product is produced only if the machine has been set up for it. Constraints (5)-(7) determine the sequence of products on the machine in each period and keep track of the machine configuration state at the beginning of each period, by recording the product that a machine is ready to process at the end of the previous one (setup carryover information is thereby tracked). Now we have a more detailed look into constraints (7). Such constraints are based on the Miller-Tucker-Zemlin sub tour elimination constraints proposed for the Traveling Salesman Problem (Laporte (1992a)).

In practice, a situation may occur where more than one lot of the same product is produced in a single period. In other words, at least one *sub tour* – a production sequence that starts and ends in the same setup state – may exist in that period.

Two special sub tour cases are referred to throughout the article: *alpha* sub tours (sub tours that start and end in the same setup state as the first setup state of each period's production sequence) and *disconnected* sub tours (sub tours that are not part of the period's main sequence). Disconnected sub tours are further classified according to their complexity: *simple* disconnected sub tours (sub tours that form a perfect loop) and *complex* disconnected sub tours (sub tours that in turn are formed by multiple sub tours).

Consider a digraph G where nodes represent production lots of product i , solid arcs (i, j) represent setups from product i to product j , and dashed arcs represent the setup states inherited from or passed to neighboring periods, thus producing a visual representation of the production sequence of a given period. Figure 1 shows some sub tour examples, including the aforementioned special cases.

It is evident that disconnected sub-tours cannot be part of a feasible solution, as it is impossible to define a finite chronological sequence that represents them correctly. The following reasoning shows that constraints (7) eliminate all but alpha sub tours.

Let $C \subseteq [N]^2$ represent a non-empty subset of product pairs (i, j) , with cardinality $|C|$. If $T_{ijt} = 1, \forall (i, j) \in C$ in a given period t , then we say C represents a sub tour (or a group of

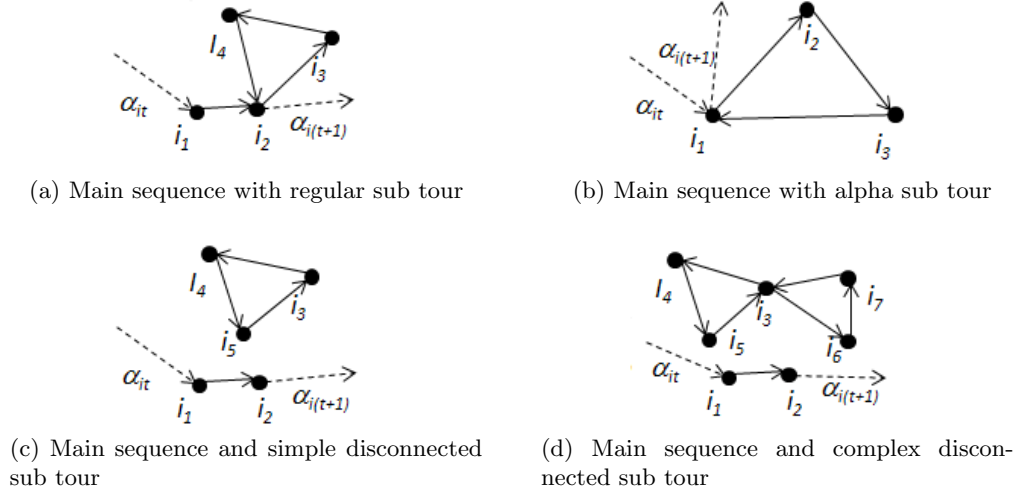


Figure 1: Sub tour examples

sub tours). By summing constraints (7) up for all $(i, j) \in C$, we get $\sum_C V_{it} + N \cdot \sum_C T_{ijt} - \sum_C (N-1) - N \cdot \sum_C \alpha_{it} \leq \sum_C V_{jt}$. Considering a *non-alpha* sub tour, we get $\sum_C \alpha_{it} = 0$. **Because** $\sum_C V_{it} = \sum_C V_{jt}$, $N \cdot \sum_C T_{ijt} = N \cdot |C|$ and $\sum_C (N-1) = N \cdot |C| - |C|$. Thus, we get $N \cdot |C| - N \cdot |C| + |C| - 0 \leq 0$, which is clearly impossible for $|C| > 0$. This guarantees no sub tour occurs.

Now we show that (7) allow *alpha* sub tours. Let C_1 denote a sub tour, such that the machine is set up for one product belonging to that sub tour, i.e. $\exists i, j : \alpha_{it} = 1 \wedge (i, j) \in C_1$. It is easy to see that such a cycle C_1 is not cut off by constraints (5)-(7).

2.2 Allowing sub tours in the main sequence

The following changes must be made to F_1 to correctly account for non-triangular setup costs and times:

Firstly, since non-triangular setup costs and times may result in a given setup being performed more than once, variables T_{ijt} must be allowed to take any non-negative integer values.

Secondly, constraints (7) must be replaced, as they only allow alpha sub tours. Let M represent a very large number, $S \subseteq [N]$ be a non-empty, non-unitary subset of the entire products set, and Y_{it} be a binary variable taking the value of 1 when the machine is configured for product i at least once in period t , and 0 otherwise:

$$Y_{it} = \begin{cases} 1 & \text{if } \sum_j T_{jit} + \alpha_{it} \geq 1, \\ 0 & \text{otherwise.} \end{cases}$$

The following constraints are valid for any feasible solution (with or without sub tours),

and cut off disconnected sub-tours:

$$\sum_{j \notin S} \sum_{i \in S} T_{jit} + \sum_{i \in S} \alpha_{it} + M \cdot \sum_{i \in S} (1 - Y_{it}) \geq 1 \quad t \in [T], S \subseteq [N], |S| \geq 2 \quad (9)$$

These constraints are non-active whenever the machine is not configured to produce at least one product i in S in period t . If this is not the case ($Y_{it} = 1$ for every $i \in S$), then (9) reduces to $\sum_{j \notin S} \sum_{i \in S} T_{jit} + \sum_{i \in S} \alpha_{it} \geq 1$. Clearly, this expression assures that the number of *inward links* (setups from another production lot, or the period's beginning) to a given set of production lots is always greater than or equal to one, as exemplified in Figure 2). If S represents a productive sub tour, it forces S 's cycle to be connected to the production sequence of the previous period (through α 's) or to the main sequence of that period (through T 's), therefore it cuts disconnected sub tours off. In other words, this requirement eliminates a disconnected multigraph on the node set S . Note that all regular sub tours (even non-alpha ones, such as S_1 on Figure 2) are allowed by (9).

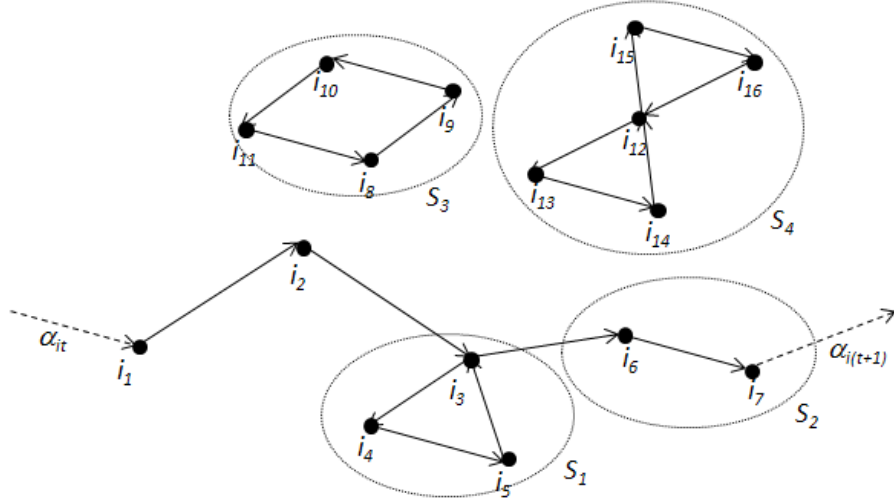


Figure 2: Examples of S : regular sub tour (S_1), part of the main sequence (no sub tour exists, S_2), simple disconnected sub tour (S_3) and complex disconnected sub tour (S_4)

The set (9) results in $T \cdot 2^N$ constraints, making full implementation impracticable. As such, individual constraints (9) will be dynamically added, as opposed to including them directly in the initial model. This can be done in a number of ways. In this article, we propose adding such constraints during the branch-and-cut process, whenever a disconnected sub tour is identified, thus removing it and preventing it thereafter. Two algorithms are used to achieve this end.

Algorithm 1, *FindST*, identifies a set of disconnected production lots S (which either form a disconnected sub tour or a group of disconnected sub tours).

```

for  $t \leftarrow 1$  to  $T$  do
  for  $i \leftarrow 1$  to  $N$  do
    if  $\alpha_{it} = 1$  then
       $Connected_{it} \leftarrow 1$ ;
       $Validate(i,t)$ ;
      break out of  $i$  loop;
    end
  end
  for  $i \leftarrow 1$  to  $N$  do
    if  $Connected_{it} = 0$  and  $Y_{it} = 1$  then a non-validated production lot exists
       $Disconnected_{it} \leftarrow 1$ ;
    end
  end
end
AddConstraints( $Disconnected$ );

```

Algorithm 1: The sub tour identification (*FindST*) algorithm

The sub tour identification algorithm is called at every feasible node during the branch-and-cut process. If any disconnected sub tour is found, a global cut will be added, eliminating and preventing it from happening again. In order to find production lots not linked to the main sequence, *FindST* algorithm calls recursive *Validate* algorithm several times (see Algorithm 2)

```

Input: Connected product  $i$  to explore
Input: Period  $t$ 
for  $j \leftarrow 1$  to  $N$  do
  if  $Connected_{jt} = 0$  then
    if  $T_{ijt} \geq 1$  then
       $Connected_{jt} \leftarrow 1$ ;
       $Validate(j,t)$ ;
    end
  end
end

```

Algorithm 2: The recursive product validation (*Validate*) algorithm

The recursive product validation algorithm works like this: Given a product i that is known to be part of the main sequence, setups from i to every other product j are checked. For each j , if at least one setup occurs from i to j ($T_{ijt} \geq 1$), and j is so far not known to be part of the main sequence, then j is *validated* as being part of the main sequence, and the product validation algorithm is called again, with j as argument. The product validation algorithm is initialized in every period with the first product in sequence as argument (given by $arg_i(\alpha_{it} = 1)$)

At the end of the process, production lots that are not validated as being part of the period's main sequence, $S_t = \{arg_i(Disconnected_{it} = 1)\}$, are known to be disconnected, and the corresponding constraint is added.

Note that this approach intends to identify and remove any disconnected sub tour en-

countered. However, it is also possible to use additional *a priori* polynomial sized constraints that prevent *simple* disconnected sub tours (thus reducing the number of dynamically added constraints) as follows:

Let a new binary variable Q_{ijt} be 1 if at least one setup operation T_{ijt} is performed, and 0 otherwise:

$$Q_{ijt} = \begin{cases} 1 & \text{if } T_{ijt} \geq 1, \\ 0 & \text{otherwise.} \end{cases}$$

The following modification of constraint (7) allows connected sub tours, while removing simple disconnected ones:

$$V_{it} + M \cdot (Q_{ijt} - 1) + M \cdot \left(Q_{ijt} - \alpha_{it} - \sum_l T_{lit} \right) \leq V_{jt} - 1 \quad i \in [N], j \in [N], t \in [T] \quad (10)$$

Constraints (10) works as follows:

Consider a general cycle $C \in [N]^2$ of size $|C|$, with $T_{ijt} \geq 1, \forall (i, j) \in C$, which may be composed of single or multiple sub tours, and let S be the node set of C . Summing all the constraints (10) up for every arc (setup) belonging to cycle C , we obtain the following requirement:

$$|C| \leq M \cdot \sum_{(i,j) \in C} \left(\alpha_{it} + \sum_l T_{lit} - 1 \right) \quad (11)$$

If cycle C corresponds to a simple disconnected sub tour, it is evident that $\sum_l T_{lit} = 1$ and $\alpha_{it} = 0, \forall i \in S$, which violates constraints (11). In any other case, $\alpha_{it} + \sum_l T_{lit} \geq 2$ for the sub tour *joint* (the product that starts and ends a regular (product i_2 in Figure 1a) or alpha (product i_1 in Figure 1b) sub tour, or bridges multiple sub tours into a complex disconnected sub tour (product i_3 in Figure 1d)), thus fulfilling the imposed requirements.

2.3 Enforcing minimum lot sizes

In cases where non-triangular inequalities exist due to the possibility to produce intermediate lower-grade or cleansing products, minimum lot sizes must be imposed, so as to guarantee an effective machine cleansing. Data representing the minimum lot sizes of each product should be added to the model:

m_i minimum size of each production lot of product i .

Additionally, a new binary variable R_t that equals to one if at least one setup is performed during period t is required:

$$R_t = \begin{cases} 1 & \text{if } \sum_i \sum_j T_{ijt} \geq 1, \\ 0 & \text{otherwise.} \end{cases}$$

The following constraints are added:

$$X_{it} = X_{it}^{-1} + X_{it}^0 \quad i \in [N], t \in [T] \quad (12)$$

$$X_{it}^{-1} \leq M \cdot \alpha_{it} \quad i \in [N], t \in [T] \quad (13)$$

$$X_{it}^0 \geq m_i \cdot \left(\sum_j T_{jit} - \alpha_{i(t+1)} \right) \quad i \in [N], t \in [T] \quad (14)$$

$$X_{it}^0 + \sum_{k=t+1}^s X_{ik}^{-1} \geq m_i \cdot \sum_j T_{jit} - M \cdot \left(\sum_{k=t+1}^{s-1} R_k + 1 - R_s \right) \quad i \in [N], t \in [T], s \in [T] \setminus [t] \quad (15)$$

Expressions (12) split production X_{it} into the amount X_{it}^0 of product i produced in period t after setups are performed in that period, and the amount X_{it}^{-1} produced at the beginning of the period, after a setup carry-over. If the setup state of product i is not carried over into period t ($\alpha_{it} = 0$), then clearly $X_{it}^{-1} = 0$, as imposed by (13). Constraints (14) assure production lots that start and end within period t fulfil the minimum lot size requirement. Constraints (15) enforce a minimum production size proportional to the number of setups to product i in period t , allowing that same production size to be split into subsequent periods. Note that this will only be enforced if there is at least one setup occurring in period s (*i.e.*, the cross over production lot ends), and there are no setups between $t+1$ and $s-1$ (*i.e.*, a unique cross over production lot is being considered). This can be simplified if production lots never span for more than one entire period (*i.e.*, $s = t+1$ instead of $s \in [T] \setminus [t]$).

Finally, variable domains must be specified:

$$(X_{it}, X_{it}^0, X_{it}^{-1}, I_{it}, \alpha_{it}, V_{it}) \geq 0, T_{ijt} \in \mathbb{N}_0, (Y_{it}, R_t) \in \{0, 1\} \quad (16)$$

The new formulation, F_2 , consists of objective function (1) subject to constraints (2)-(6), (9) and (12)-(16). We prove in the following lemma that F_1 is a special case of F_2 , and, as such, the optimal solution of F_2 is at least as good as F_1 's:

Lemma 1. $v(F_1) \geq v(F_2)$

Proof. F_2 can be seen as a generalization of F_1 , since the latter can be derived from the former by adding additional constraints. If minimum lot sizes are not enforced, by setting $m_i = 0$, $\forall i \in [N]$, then constraints (12)-(15) become redundant and can be dropped. In addition, let the set

$$\sum_{(i,j) \in C} \alpha_{it} \geq \frac{|C| \cdot (1 - N)}{N} + \sum_{(i,j) \in C} T_{ijt} \quad (17)$$

be added to F_2 , obtained by all the constraints (7) listed for every arc (i, j) of a general cycle C . As $\sum_{(i,j) \in C} T_{ijt} \geq |C|$, $\frac{|C|}{N} > 0$ and α_{it} can only take on integer values, then (17) reduces to $\sum_{(i,j) \in C} \alpha_{it} \geq 1$. If S contains the node set of cycle C , then (9) is equivalent to $\sum_{i \in S} \alpha_{it} \geq 1 - \sum_{j \notin S} \sum_{i \in S} T_{jit}$. Clearly, (17) makes this constraint redundant as $\sum_{i \in S} \alpha_{it} =$

1 if $\sum_{(i,j) \in C} \alpha_{it} \geq 1$. In the presence of sub tours, (17) dominates (9), making the feasible solutions of F_2 with (17) coincident with those of F_1 . Therefore, the set of feasible solutions of F_1 is a subset of the set of feasible solutions of F_2 . Consequently, for the same data set, $v(F_1) \geq v(F_2)$, completing the proof. \square

The following example demonstrates the previous statement by showing the optimal solutions of the same instance to F_1 and F_2

Example 1. A production plan of five different products, $i = \{1, 2, 3, 4, 5\}$, over the next three periods must be devised. A certain component of product 5 contaminates product 1, and an expensive disinfectant product is required to clean the machine, thus increasing c_{51} . Product 3 has a component that absorbs the contaminating component from product 5, hence the triangular inequality will not hold for the setup costs of sequence 5–3–1. Table 1 shows the relevant data for this problem. Additionally, consider $c_{ij} = 10s_{ij}, \forall (i, j) \setminus (5, 1)$, $c_{51} = 250$, and $C_t = 100, \forall t$.

Table 1: Data for the five product, three period problem

	d_{it}			s_{ij}					h_i
	$t = 1$	$t = 2$	$t = 3$	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	
$i = 1$	90	0	110	-	20	100	100	100	10
$i = 2$	0	10	0	100	-	5	100	100	10
$i = 3$	0	10	0	10	100	-	10	100	10
$i = 4$	0	10	0	100	100	100	-	10	10
$i = 5$	0	10	0	10	100	5	100	-	10

Tables 2 and 3 show the most relevant non-zero solution values given by F_1 and F_2 , respectively. Those same solutions are graphically represented by figures 3 and 4. Here, white blocks represent production that is to be consumed in that period, light grey blocks represent production that is to be stocked, middle grey represents idle time and dark grey represents setups.

Table 2: F_1 's optimal solution

$t = 1$	$t = 2$	$t = 3$
$\alpha_{11} = 1$	$\alpha_{12} = 1$	$\alpha_{13} = 1$
$X_{11} = 95$	$X_{12} = 5, T_{122} = 1$	$X_{13} = 100$
	$X_{22} = 10, T_{232} = 1$	
	$X_{32} = 10, T_{342} = 1$	
	$X_{42} = 10, T_{452} = 1$	
	$X_{52} = 10, T_{512} = 1$	
$I_{11} = 5$	$I_{12} = 10$	

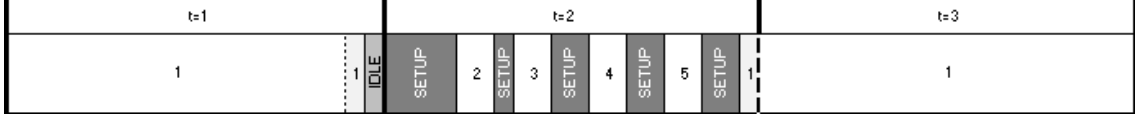


Figure 3: Graphical representation of F_1 's optimal solution

Note that in F_1 's optimal solution, 5 units of product 1 must be added to stock in periods 1 and 2 to fulfill demand in period 3, with a holding cost of 150 monetary units. Setup costs account for 700 monetary units, resulting in an objective function value of 850 monetary units.

Table 3: F_2 's optimal solution

$t = 1$	$t = 2$	$t = 3$
$\alpha_{11} = 1$	$\alpha_{12} = 1$	$\alpha_{13} = 1$
$X_{11} = 100$	$T_{122} = 1$	$X_{13} = 100$
	$X_{22} = 10, T_{232} = 1$	
	$X_{32} = 10, T_{342} = 1, T_{312} = 1$	
	$X_{42} = 10, T_{452} = 1$	
	$X_{52} = 10, T_{532} = 1$	
$I_{11} = 10$	$I_{12} = 10$	

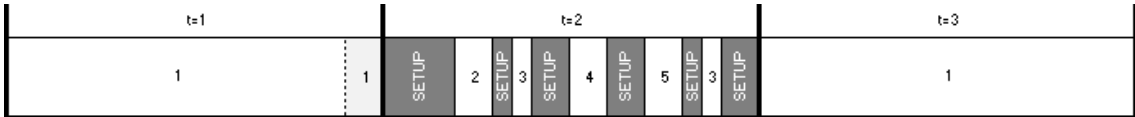


Figure 4: Graphical representation of F_2 's optimal solution

In F_2 's optimal solution, an extra setup exists. However, due to the non triangular inequality of sequence 5 – 3 – 1, total setup costs are reduced by 100 monetary units, to a total cost of 600. The increase in setup times forces production of product 1 in period 2 (see Figure 3) to be moved to period 1, increasing holding costs by 50 monetary units, to a total of 200. This results in an objective function value of 800 monetary units, which is 50 less than F_1 's. Note that our solution does not include the size of each individual production lot, but instead the total amount of each product to be produced in each period. The example depicted in Figure 4 represents one of the many possible ways to split $X_{32} = 10$ units of product 3 between the two corresponding production lots in period 2. These variations have no impact in the objective function value.

2.4 General Lotsizing and Scheduling Problem

The class of small-bucket lot-sizing and scheduling problems tries to capture both lot-sizing and scheduling decisions, Drexl and Kimms (1997). Wolsey (2002) provides a comprehensive study and classifications scheme for different small-bucket and big-bucket models. His analysis shows that the LP-relaxation of small-bucket models usually delivers very weak lower bounds. Only with customized reformulations and valid inequalities added to the problem is an improvement of the lower bound possible. In contrast most big-bucket models provide much better lower bounds.

Fleischmann and Meyr (1997) develop a model based on a mixed small-bucket-big-bucket general lotsizing and scheduling model (GLSP), where each period of the planning horizon is divided into a fixed number $|S_t|$ of micro-periods of variable length, representing the maximum number of lots that can be scheduled in (macro-)period t . Here the production sequence within each period is explicitly obtained by assigning an item to each micro-period. Let the new variable Z_{ijs} indicate whether or not a changeover occurs on the machine configuration state from product i to j in micro-period s . The GLSP is formulated as:

$$v(F_{GLSP}) = \min \sum_i \sum_j \sum_s c_{ij} \cdot Z_{ijs} + \sum_i \sum_t h_i \cdot I_{it} \quad (18)$$

$$I_{it} = I_{i(t-1)} + \sum_{s \in S_t} X_{is} - d_{it} \quad i \in [N], t \in [T] \quad (19)$$

$$\sum_i \sum_{s \in S_t} p_i \cdot X_{is} + \sum_i \sum_j s_{ij} \cdot Z_{ijs} \leq C_t \quad t \in [T] \quad (20)$$

$$X_{is} \leq \frac{C_t}{p_i} \cdot Y_{is} \quad i \in [N], s \in [S] \quad (21)$$

$$X_{is} \geq m_i \cdot (Y_{is} - Y_{i(s-1)}) \quad i \in [N], s \in [S] \quad (22)$$

$$\sum_i Y_{is} = 1 \quad s \in [S] \quad (23)$$

$$Z_{ijs} \geq Y_{i(s-1)} + Y_{js} - 1 \quad i, j \in [N], s \in [S] \quad (24)$$

$$(X_{is}, I_{it}, Z_{ijs}) \geq 0, Y_{is} \in \{0, 1\}. \quad (25)$$

(18) expresses the inventory holding costs and setup costs. Demand is met without backloging by (19). Capacity constraints are given in (20). Note that the length of micro-period s is not constant but rather is determined by the capacity consumption of the (only) product that is setup and produced within it. Constraint (21) links production and setup variables. Minimum lot-sizes are ensured by requirements (22). Furthermore, at most one product can be produced per micro-period (23). Constraint (24) forces a changeover to be performed if two different products are set up in two consecutive micro-periods.

In case the triangular inequality does not hold, GLSP enables an item to be produced several times in the same macro-period. Despite allowing for a very accurate modeling of the situation, this model is computationally very hard to solve, whereas the big-bucket model introduced in Section 2.2 is much easier to tackle, as shown below.

2.5 Comparison of CLSD-like model with GLSP

Here our aim is to compare the efficiency of the CLSD-like model and GLSP when solved by a search engine. Both formulations have been tested on well known instances reported in Fleischmann (1994) with eight products and eight periods. Problems TV11, TV13 and TV14 differ only in the capacity of the macro-periods and have a capacity utilization (defined as $\sum_i d_{it}/C_t$) per period of 97%, 76% and 64%, respectively. Fleischmann (1994) relies on various setup cost matrices (c_{ij}). We present results for S1, S2, S3 and S4. The values of S1 and S3 are generated at random from the intervals [0,600] and [0,300], respectively. The entries of S2 are taken at random from the values 0,100,200,...,600. The S4 matrix represents a typical situation in practice with only two kind of setups, a major setup cost (cost 500 between product families) and a minor setup cost (cost 100 within the same family). We note that matrix S4 satisfies the triangle inequality, while all the other matrices do not. These instances are available online in Instances (2009). Each macro-period has been subdivided into 8 and 10 GLSP micro-periods.

All the computational results reported have been obtained by running the commercial solver CPLEX 12.1 from ILOG, on a Pentium T7700 CPU running at 2.4 GHz with 4GB of random access memory. The maximum time for the search overall was set at one hour. The algorithms 1 and 2 to solve F_2 were coded in C++ using Concert 2.7.

In presenting the results in Table 4, we list the problem type, the optimal values provided by formulations F_1 and F_2 , as well as the respective solution times (in seconds), and the percentage improvement $\text{gap}()$ of $v(F_2)$ over $v(F_1)$. The last columns of the table report the upper (UB) and lower (LB) bounds to F_{GLSP} , considering eight and ten micro-periods, for one hour-time limit.

Table 4: Computational Results

Prob.	set. cost matrix	$v(F_1)$	time(s)	$v(F_2)$	time(s)	gap()	GLSP N = 8			GLSP N = 10		
							UB	LB	time(s)	UB	LB	time(s)
TV11	S1	2640	31	2412	17	9.5%	3122	47	3600	2849	2.7	3600
TV11	S2	1285	33	1239	43	3.7%	1285	107	3600	1239	31.4	3600
TV11	S3	1413	45	1308	21	8.0%	1816	11	3600	1566	1.6	3600
TV11	S4	6417	2	6417	1	0.0%	8060	72	3600	7636	0.2	3600
TV13	S1	2291	10	2180	4	5.1%	2461	19	3600	2574	0.1	3600
TV13	S2	1150	51	1140	188	0.9%	1170	11	3600	1183	0.0	3600
TV13	S3	1317	10	1259	20	4.6%	1546	9	3600	1409	0.1	3600
TV13	S4	4960	4	4960	3	0.0%	6090	19	3600	5783	0.0	3600
TV14	S1	2188	3	2180	7	0.4%	2475	33	3600	2485	0.1	3600
TV14	S2	1150	53	1140	256	0.9%	1360	6	3600	1150	0.3	3600
TV14	S3	1307	10	1259	21	3.8%	1337	12	3600	1364	0.1	3600
TV14	S4	4920	5	4920	8	0.0%	5430	13	3600	5470	0.4	3600

In all instances except those with setup cost matrix S4, the formulation that allows for subtours in the main sequence (F_2) has computed a solution that is less costly (with a statistically significant p-value of 0.02) than the corresponding F_1 solution, which only

incorporates alpha-subtours. The solutions have the same cost in the case of setup cost matrix $S4$. The difference increases with increasing capacity utilization (e.g. compare instance TV11/S1 to TV13/S1 and TV14/S1). The respective solution times are not significantly different (p-value = 0.19). Since $S4$ obeys the triangular inequality, there is no advantage in having non-alpha subtours in the production sequence.

Compared to the solutions given by small-bucket models, it is clear that F_2 is always more efficient than F_{GLSP} . Model F_{GLSP} is not able to solve to optimality within one hour any instance type, except for instance TV11/S2 with 10 micro-periods, but CPLEX does not prove optimality due to very weak lower bound. Considering this time limit, it cannot be stated that the quality of the UB increases with the number of micro-periods, whereas the LB clearly decreases.

3 New Model for CLSD with Sequence-dependent and Period Overlapping Setup Costs and Times

Both F_1 and F_2 only take into account solutions that entail setups performed entirely within a time period. We now consider lotsizing and scheduling problems where setups are allowed to overlap period's boundaries. Such feature is of utmost importance to tackle tight capacity scenarios.

Our model uses two new types of variables (one of them binary) in addition to the variables from model F_2 . Continuous variables S_t contain the amount of time still needed to finish the last setup operation at the end of period t (cross over time). Binary variables B_{ijt} indicate whether or not the cross over setup from period t to period $t + 1$ is from product i to j .

Due to setup cross overs, setup times S_t that are delayed to the following periods (as well as setup times S_{t-1} that are inherited from previous ones) must be taken into account. Thus, capacity constraints (3) must be extended in the following way:

$$\sum_i p_i \cdot X_{it} + \sum_i \sum_j s_{ij} \cdot T_{ijt} - S_t + S_{t-1} \leq C_t \quad t \in [T] \quad (26)$$

To ensure setup cross overs S_t only occur if a given setup T_{ijt} is performed and that they do not exceed the corresponding setup time s_{ij} we add:

$$S_t \leq \sum_i \sum_j s_{ij} \cdot B_{ijt} \quad t \in [T] \quad (27)$$

$$B_{ijt} \leq Q_{ijt} \quad i \in [N], j \in [N], t \in [T] \quad (28)$$

To ensure only the last setup performed may cross over, we add:

$$\sum_j B_{jit} \leq \alpha_{i(t+1)} \quad i \in [N], t \in [T] \quad (29)$$

Note that since $\sum_i \alpha_{it} = 1$, we have $\sum_i \sum_j B_{ijt} \leq 1$, which prevents multiple setups from crossing over.

Constraints (4) must be extended so that production of product i in period t may only occur if at least one full setup operation for that product ends in that period.

$$X_{it} \leq M_{it} \cdot \left(\sum_j (T_{jit} - B_{jit}) + \alpha_{it} \right) \quad i \in [N], t \in [T] \quad (30)$$

Note that this constraint may become very loose, but capacity constraints and inventory costs will always prevent X_{it} from getting too big.

Minimum lot sizes must be enforced when a setup crosses over:

$$\sum_{k=t+1}^s X_{ik}^{-1} \geq m_i \cdot \sum_j B_{jit} - M \cdot \left(\sum_{k=t+1}^{s-1} R_k + 1 - R_s \right) \quad i \in [N], t \in [T], s \in [T], s > t \quad (31)$$

Finally, we assure S_t is non-negative, and Q_{ijt} and B_{ijt} are binary:

$$S_t \geq 0, (Q_{ijt}, B_{ijt}) \in \{0, 1\}. \quad (32)$$

The new formulation, F_3 , consists of objective function (1) subject to constraints (2), (5), (6), (9), and (12)–(32).

Let S_2 and S_3 be the sets of feasible solutions to F_2 and F_3 , respectively. We prove in the following lemma that F_2 is a special case of F_3 .

Lemma 2. $S_2 \subseteq S_3$.

Proof. Let us assume another model and its feasible solution set, F_3^* and S_3^* , respectively, similar to F_3 with the following additional requirement:

$$\sum_i \sum_j \sum_t B_{ijt} = 0 \quad (33)$$

This constraint ensures no setup cross over occurs. Constraints (27) make variables S_t all equal to 0. It becomes obvious that F_3^* is equivalent to F_2 , and therefore $S_2 = S_3^*$. Since F_3^* is a restricted version of F_3 , we can conclude that F_3 admits all of F_3^* 's feasible solutions, *i.e.*, $S_3^* \subseteq S_3$, which is equivalent to $S_2 \subseteq S_3$. \square

The following example demonstrates that F_3 can achieve a better optimal solution than F_2 :

Example 2. Consider the same data set of Example 1. Figure 5 shows the optimal solutions to F_3 .

Table 5: F_3 's optimal solution

$t = 1$	$t = 2$	$t = 3$
$\alpha_{11} = 1$	$\alpha_{22} = 1$	$\alpha_{13} = 1$
$X_{11} = 90, T_{121} = 1$	$X_{12} = 10$	$X_{13} = 100$
$B_{121} = 1, S_1 = 10$	$X_{22} = 10, T_{232} = 1$	
	$X_{32} = 10, T_{342} = 1, T_{312} = 1$	
	$X_{42} = 10, T_{452} = 1$	
	$X_{52} = 10, T_{532} = 1$	
	$I_{12} = 10$	

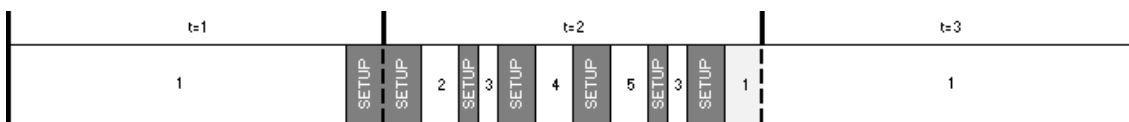


Figure 5: Graphical representation of F_3 's optimal solution

By allowing the first setup to cross over, it is possible that the extra 10 (inventory) units of product 1 that are being produced in period 1 in F_2 's optimal solution are pushed into period 2, reducing holding costs by 100 monetary units, while keeping the same setup costs. This results in an objective function value of 700 monetary units, which is 100 less than F_2 's.

4 Concluding remarks

In this paper, we have presented a novel formulation for the capacitated lotsizing and scheduling problem which correctly handles non-triangular setup costs and times while enforcing the necessary feature of minimum lot size, and allows setup cross overs between adjacent periods. **These extensions** open ways for solutions not being considered so far, which may lead to improvements in overall production planning efficiency, and reduce the total cost of production plans.

Additionally, we have developed a method for dynamically identifying and removing disconnected sub tours. Such a method is required for large problems, since the direct implementation of our model would require the use of an exponential number of constraints. The simplicity of this method makes it easy to implement in most programming languages usually combined with optimization engines. We prove that our big-bucket approach performs computationally much better than other small-bucket treatments. Nevertheless, an important future research question is to find a polynomial sized set of constraints that cut disconnected sub tours off, while enabling all types of connected cycles.

Extensions to this model considering multiple machines, shortages, backlogging costs and maximum lot sizes are straightforward, making it a good starting point for models reflecting a wide range of real life situations.

Acknowledgments

This research was partially funded by FP7-PEOPLE-2009-IRSES project, no. 246881.

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.
- E. Balas. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636, 1989.
- 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 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*, 19:11–21, 1997.
- M. Gopalakrishnan. A modified framework for modelling set-up carryover in the capacitated lotsizing problem. *International Journal of Production Research*, 38(14):3421–3424, 2000.
- M. Gopalakrishnan, D. M. Miller, and C. P. Schmidt. A framework for modelling setup carryover in the capacitated lot sizing problem. *International Journal of Production Research*, 33(7):1973–1988, 1995.
- K. Haase. Capacitated lot-sizing with sequence dependent setup costs. *OR Spektrum*, 18: 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):159–169, 2000.
- Instances, 2009. URL <http://paginas.fe.up.pt/balobo/NonTriangularSetups>.
- G. Laporte. The travelling salesman problem: An overview of exact and approximated algorithms. *European Journal of Operational Research*, 59(2):231–248, 1992a.
- G. Laporte. The vehicle routing problem: An overview of exact and approximated algorithms. *European Journal of Operational Research*, 59(2):345–358, 1992b.

- P. Porkka and A. P. J. V. M. Kuula. A modified framework for modelling set-up carryover in the capacitated lotsizing problem. *International Journal of Production Research*, 38(14):3421–3424, 2000.
- C. R. Sox and Y. Gao. The capacitated lot sizing problem with setup carry-over. *IIE Transactions*, 31(2):173–181, 1999.
- C. Suerie. Modeling of period overlapping setup times. *European Journal of Operational Research*, 174(2):874–886, 2006.
- C. Suerie and H. Stadtler. The capacitated lot-sizing problem with linked lot sizes. *Management Science*, 49(8):1039–1054, 2003.
- C. Sung and C. T. Maravelias. A mixed-integer programming formulation for the general capacitated lot-sizing problem. *Computers & Chemical Engineering*, 32:244–259, 2008.
- E. A. V. Toso and R. Morabito. Otimização no dimensionamento e seqenciamento de lotes de produção: Estudo de caso numa fábrica de rações. *Gestão e Produção*, 12(2):203–217, 2005.
- L. Wolsey. Solving multi-item lot-sizing problems with an mip solver using classification and reformulation. *Management Science*, 48(12):1587–1602, 2002.
- X. Y. Zhu and W. E. Wilhelm. Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE Transactions*, 38(11):987–1007, 2006.