Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

An analysis of formulations for the capacitated lot sizing problem with setup crossover

Diego Jacinto Fiorotto [a,*], Raf Jans [b], Silvio Alexandre de Araujo [a]

[a] *Departamento de Matemática Aplicada, Univ. Estadual Paulista, 15054-000 São José do Rio Preto, SP, Brazil*
[b] *HEC Montréal and CIRRELT, H3T 2A7 QC, Canada*

ARTICLE INFO

Article history:
Received 6 November 2015
Received in revised form 21 December 2016
Accepted 27 December 2016
Available online 1 January 2017

Keywords:
Production
Mathematical formulations
Lot sizing
Setup crossover
Symmetry breaking

ABSTRACT

The lot sizing problem with setup crossover is an extension of the standard big bucket capacitated lot sizing problem (CLSP). The general idea is that the first setup operation of each planning period can already start in the previous period, if not all the capacity is used in that previous period. This provides more flexibility in the planning and increases the possibility of finding feasible and better solutions compared to the standard assumption. Two different formulations have been presented in the literature to model a setup crossover. Since these formulations have not been compared directly to each other, we present a computational study to determine which is the best formulation. Furthermore, we explore ideas indicating that in one of the formulations from the literature it is not necessary to impose binary conditions on the crossover variables and we propose symmetry breaking constraints for both formulations from the literature. Finally, we quantify the value of this type of flexibility in a computational experiment and analyze which factors influence this value.

1. Introduction

The research on dynamic lot sizing in discrete time started over 50 years ago with the seminal papers of Wagner and Whitin (1958) and Manne (1958). Over the past decades, there has been an increasing interest in the application of these models, and researchers have been able to incorporate more and more real world features into lot sizing problems.

The lot sizing problem is a production optimization problem which involves determining how many items to produce in each period in order to meet the demand for these items. The resulting production plan should minimize the sum of the setup, production and inventory holding costs. The problem considered in this work is the single stage, single machine, multi-product, big time bucket lot sizing problem with setup times. Several different products can be produced in the same time period on the same machine. A setup must be done for each type of product that is produced in a specific period. In the standard version of this problem the setup for the first product type produced in a period starts at the beginning of that period (Trigeiro, Thomas, & McClain, 1989). In this paper we study an extension of this lot sizing problem that includes the possibility of a setup crossover. The idea is that in certain cases setup operations can be interrupted at the end of a period and resumed at the beginning of the next period, in other words, the setups can span over two periods. This implies that the first setup in period t can already start at the end of period $t-1$ if there is some capacity left, and continue at the beginning of period t (Menezes, Clark, & Almada-Lobo, 2010). This flexibility can result in more efficient solutions compared to the standard assumption (where the setup time is restricted to be contained within the period) since we free up capacity in period t by (partially) moving the setup of the first product to the previous period. In the big bucket models, the setup times are smaller than the capacity limit.

Typically, the setups include machine adjustments, calibration, inspection and cleaning activities that are required before switching over the resource to produce another product. Quite often, setup operations can be interrupted at the end of a period (e.g. just before the weekend break) and resumed at the beginning of the next one (e.g. just after the weekend break). In other cases, the operation is run continuously, and there is no period of interruption between the end of one period and the start of the next one. In both cases, the setup can be split between two periods. We give some examples. In the beverage industry, the setup of the beverage production line consists in preparing the syrup in tanks and distribute it to parallel bottling machines. In some cases this process can be interrupted and resumed the next period. In the fabrication of steel components, different molds are needed to produce differ-

[*] Corresponding author.
E-mail addresses: diego_fiorotto@homail.com (D.J. Fiorotto), raf.jans@hec.ca (R. Jans), saraujo@ibilce.unesp.br (S.A. de Araujo).

http://dx.doi.org/10.1016/j.cie.2016.12.037
0360-8352/© 2016 Published by Elsevier Ltd.

ent products. The setup consists of changing the mold and in some cases this process can be interrupted and resumed without any problem.

It is important to note the differences between the concepts of setup crossover and setup carryover. While with setup crossover the setups can span over two periods, the setup carryover allows a setup state to be maintained from one period to the next one. In other words, if we finish a period t producing a particular item i it is possible to start producing the item i in period $t+1$ without performing a new setup for this item.

Although setup crossover is a natural extension of the standard assumption, just a few studies have considered it, due to the difficulty in dealing with the underlying problems (Mohan, Gopalakrishnan, Marathe, & Rajan, 2012; Belo-Filho, Almada-Lobo, & Toledo, 2014). All the studies that handle setup crossovers in their formulations have added extra binary variables to the formulations indicating if there is a setup crossover in a period or not, which increases the difficulty of the formulations.

The aim of this paper is: (1) to compare the two formulations proposed in the literature to determine which formulation is the best; (2) to propose new constraints to break the symmetry which is present in the formulations from the literature; (3) to prove that in one of the formulations from the literature we do not need binary conditions on the crossover variables; (4) to analyze the impact of the proposed adaptations of these formulations (i.e. no binary variables and symmetry breaking constraints) in computational experiments, and (5) to determine the value of the flexibility provided by the setup crossover and analyze the factors that have an impact on this value.

We also have explored other ideas to avoid the necessity of defining new extra binary variables to model the setup crossover. Two new formulations were proposed and can be found in a technical report (Fiorotto, Jans, & de Araujo, 2014) which includes some theoretical and computational results. These two formulations present more restricted models, and hence provide only an upper bound on the optimal solution for the model with setup crossover. The computational experiments indicated that these two restricted formulations without extra binary variables for the setup crossover actually take substantially more time to be solved compared to the best formulation for the setup crossover. Therefore, these two restricted formulations are not included in this paper.

The paper is organized as follows. In Section 2, we provide a literature review on lot sizing problems with setup crossover. Section 3 presents the formulations from the literature along with the new proposed formulations including some theoretical results for the formulations. Section 4 describes the computational results and analyses and finally in Section 5, we present our conclusions.

2. Literature review

There is a vast amount of literature on big-bucket capacitated lot sizing problems (CLSP) with setup times, where setup times have to be contained completely within one period (Trigeiro et al., 1989). These models have been extended to deal with various industrial issues (see Jans & Degraeve (2008) for an overview), including setup carryover and setup crossover.

Several papers analyze the extension with setup carryover. Sox and Gao (1999) propose two formulations for the CLSP with setup carryover. The first one extends the formulation proposed by Trigeiro et al. (1989) and the second one uses the shortest path reformulation and the ideas proposed by Eppen and Martin (1987). Suerie and Stadtler (2003) propose a formulation for the CLSP with setup carryover based on the simple plant location formulation (Krarup & Bilde, 1977) and their computational tests have shown that this formulation is better than the formulations proposed by Sox and Gao (1999). Gopalakrishnan, Ding, Bourjolly, and Mohan (2001) develop a tabu search heuristic to solve the CLSP with setup carryover and using the data sets from Trigeiro et al. (1989) they compute the effectiveness of the setup carryover. Their results indicate an 8% reduction in total cost on average through setup carryover compared with the standard CLSP.

Regarding the problem with setup crossover for the small bucket problem, Suerie (2006) studies the lot sizing and scheduling problem and proposes two formulations that correctly handle setup crossovers which allow "long" setup times (i.e. setup times can be bigger than the capacity in one period). The author compares his results with the results found by the standard lot sizing and scheduling problem and concludes that the proposed formulations remove infeasibility and produce improved solutions in certain cases.

For the big bucket problem, Sung and Maravelias (2008) present a mixed-integer programming formulation for the capacitated lot sizing problem allowing setup carryover and crossover (CLSP-SCC). The authors consider sequence independent setups, non-uniform time periods and long setup times. They show in a detailed way how to deal with the boundary of the periods using setup crossover with the assumption that the setup cost is accounted for at the beginning of the setup. Finally they discuss how their formulation can be extended for problems with idle time, parallel units, families of products, backlog and lost sales.

Menezes et al. (2010) propose a formulation for the CLSP-SCC considering sequence-dependent and non-triangular setups, allowing subtours and enforcing minimum lot sizing. They propose two lemmas to demonstrate that their formulation is more efficient than the classical lot sizing and scheduling problem. Moreover, they present an example that shows the improvement of the solutions allowing setup crossover compared to the classical formulation.

Kopanos, Puigjaner, and Maravelias (2011) develop a formulation for the CLSP-SCC with backlog where the items are classified into families. The approach considers that the setups are family sequence-dependent, and sequence-independent for items belonging to the same family. The formulation is tested for a complex real world problem in the continuous bottling stage of a beer production facility and it finds good solutions for problems with hundreds of items.

Mohan et al. (2012) include the possibility of setup crossover for the formulation proposed by Suerie and Stadtler (2003) that handles the problem with setup carryover and compare the improvement obtained by adding the crossover in the formulation with setup carryover. They conclude that in nine out of fifteen problem instances tested, their formulation yielded better solutions or removed infeasibility.

Camargo, Toledo, and Almada-Lobo (2012) propose three formulations for the two-stage lot sizing and scheduling problem and one of these considers setup crossover, which is achieved by a continuous-time representation. From the computational results, they conclude that despite delivering the worst performance in terms of CPU times, the formulation with setup crossover is the most flexible of the three to incorporate setup-related features.

Belo-Filho et al. (2014) consider the problem CLSP-SCC with backlog. They propose two formulations for the problem, the first one is built on top of the formulation of Sung and Maravelias (2008) and the second one proposes a time index disaggregation, defining the start and the completion time periods of the setup operation. They show the relationship between the proposed formulations and compare their formulations with the formulation proposed by Sung and Maravelias (2008). Finally they point out that setup crossover is an important modeling feature in case setup times consume a considerable part of the period capacity.

3. Mathematical models

In this section, we first present the classical formulation (without crossover) using the simple plant location reformulation (Krarup & Bilde, 1977). Next, we present two formulations for the problem with setup crossover based on the ideas proposed by Menezes et al. (2010) and Mohan et al. (2012). We further prove that the binary conditions on the new crossover variables can be omitted in the formulation proposed by Menezes et al. (2010). We also propose new constraints for the formulations proposed by Menezes et al. (2010) and Mohan et al. (2012) to break the symmetry resulting from the presence of alternative optimal solutions. Finally, we present a small example for which we show the solutions obtained by the studied formulations.

3.1. Classical model

Various research papers have used alternative formulations to model the classical lot sizing formulation. Two important reformulations have been proposed. A first one deals with the reformulation of the problem as a Shortest Path problem in which a redefinition of the variables proposed by Eppen and Martin (1987) is used (e.g. Fiorotto, A, & Araujo (2014)). A second one consists of a reformulation based on the Simple Plant Location problem studied in Krarup and Bilde (1977). Various theoretical and computational results concerning such reformulations have been published in the literature (see for example, de Araujo, de Reyck, Degraeve, Fragkos, & Jans (2015)). Considering that the linear relaxations of these alternative formulations are stronger than of the classical formulation, and after performing some preliminary computational tests we have chosen to use the simple plant location reformulation for all formulations presented in this paper. See for example Trigeiro et al. (1989) and Jans and Degraeve (2007) for the regular formulation.

The parameters and variables used in the formulations are described as follows:

Parameters	
$I = \{1, \dots, n\}$	set of items;
$T = \{1, \dots, m\}$	set of periods;
d_{it}	demand of item i in period t;
hc_{it}	unit inventory cost of item i in period t;
sc_{it}	setup cost for item i in period t;
vc_{it}	production cost of item i in period t;
st_{it}	setup time for item i in period t;
vt_{it}	production time of item i in period t;
Cap_t	capacity (in units of time) in period t;
cs_{itk}	total production and holding cost for producing item i in period t to satisfy the entire demand of period k, $cs_{itk} = (vc_{it} + \sum_{u=t}^{k-1} hc_{iu})d_{ik}$.
Decision variables	
x_{itk}	fraction of the demand for item i in period k produced in period t;
y_{it}	binary setup variable, indicating the production or not of item i in period t;

- Simple plant location reformulation ($F0$)

$$v(F0) = Min\left(\sum_{i=1}^{n}\sum_{t=1}^{m} sc_{it}y_{it} + \sum_{i=1}^{n}\sum_{t=1}^{m}\sum_{k=t}^{m} cs_{itk}x_{itk}\right) \tag{1}$$

Subject to:

$$\sum_{k=1}^{t} x_{ikt} = 1 \quad \forall i \in I, t \in T | d_{it} > 0 \tag{2}$$

$$\sum_{i=1}^{n} st_{it}y_{it} + \sum_{i=1}^{n}\sum_{k=t}^{m} vt_{it}d_{ik}x_{itk} \leqslant Cap_t \quad \forall t \in T \tag{3}$$

$$x_{itk} \leqslant y_{it} \quad \forall i \in I, t \in T, k \in T, k \geqslant t \tag{4}$$

$$y_{it} \in \{0,1\}, x_{itk} \geqslant 0 \quad \forall i \in I, t \in T, k \in T, k \geqslant t \tag{5}$$

The objective function (1) minimizes the total cost, which consists of the setup cost and the aggregated production and holding costs. The constraints (2) ensure that demand is met for each period. The capacity constraints (3) limit the sum of the total setup and production times. The setup constraints (4) do not allow any production in period t unless a setup is done. Finally, constraints (5) define the variable domains.

3.2. Models proposed in literature for the problem with setup crossover

In this section we present the formulations of the lot sizing problem with setup crossover proposed in the literature. These formulations are based on the formulations of Menezes et al. (2010) and Mohan et al. (2012). Both papers also include the possibility of setup carryover. We present here the way the setup crossover is formulated in these papers, without considering the setup carryover extensions. There are others papers in the literature for extensions of the CLSP with setup crossover as discussed in the literature review. However, for these formulations the way to model the setup crossover is similar to that of the papers previously mentioned, and therefore they will be omitted.

3.2.1. Model Proposed by Menezes et al. (2010) ($F1$)

Before presenting the formulation, we need to define some new variables:

New decision variables

v_{it}:	binary variable, indicating if the setup is split between period t and period $t+1$ for item i;
u_t:	extra time borrowed in period t for the setup in period $t+1$.

We assume that when a setup is split between periods t-1 and t (i.e. $v_{i,t-1} = 1$), then the setup (cost and time) is associated with the period in which the actual production is done (i.e. the second period of the cross-over). The first mathematical formulation ($F1$) based on the ideas proposed by Menezes et al. (2010) is as follows:

$$v(F1) = Min\left(\sum_{i=1}^{n}\sum_{t=1}^{m} sc_{it}y_{it} + \sum_{i=1}^{n}\sum_{t=1}^{m}\sum_{k=t}^{m} cs_{itk}x_{itk}\right) \tag{6}$$

Subject to:

$$\sum_{k=1}^{t} x_{ikt} = 1 \quad \forall i \in I, t \in T | d_{it} > 0 \tag{7}$$

$$\sum_{i=1}^{n} st_{it}y_{it} + \sum_{i=1}^{n}\sum_{k=t}^{m} vt_{it}d_{ik}x_{itk} + u_t \leqslant Cap_t + u_{t-1} \quad \forall t \in T \tag{8}$$

$$x_{itk} \leqslant y_{it} \quad \forall i \in I, t \in T, k \in T, k \geqslant t \tag{9}$$

$$u_{t-1} \leqslant \sum_{i=1}^{n} v_{i,t-1} st_{it} \quad \forall t \in T \tag{10}$$

$$v_{i,t-1} \leqslant y_{it} \quad \forall i \in I, t \in T \tag{11}$$

$$\sum_{i=1}^{n} v_{i,t-1} \leqslant 1 \quad \forall t \in T \tag{12}$$

$$y_{it} \in \{0,1\}, v_{i,t-1} \in \{0,1\}, v_{i0} = 0, u_{t-1} \geqslant 0, u_0 = 0 \quad \forall i \in I, t \in T \tag{13}$$

$$x_{itk} \geqslant 0 \quad \forall i \in I, t \in T, k \in T, k \geqslant t \tag{14}$$

The objective function (6) minimizes the total cost. Constraints (7) guarantee that the demand is satisfied in each period. The capacity constraints (8) limit the sum of the total setup times and production times, considering the time borrowed from the previous period and the time added to the next period in case of setup crossover. The setup constraints (9) do not allow any production in period t unless a setup is completed in period t. Constraint (10) limits the borrowed time in period $t-1$ to be used in period t to the value of the setup time of the product for which we allow the crossover. We cannot have a crossover from period $t-1$ to period t if there is no setup in period t, as imposed by constraint (11). Constraints (12) state that the setup can be split across periods for at most one item and finally, the conditions (13) and (14) on the variables complete the formulation.

3.2.2. Model Proposed by Mohan et al. (2012) (F2)

The second formulation is based on the ideas proposed by Mohan et al. (2012) and the main difference is the way to limit the time for the setup crossover (constraints (10) and (11) of the previous formulation). Before presenting the formulation, we need to define other new variables:

New decision variables:

z_{it}: = 1 if a complete setup is done in period t for item i, 0 otherwise;
w_{it}: = 1 if there is a setup crossover between period $t-1$ and period t for item i with splits of the setup time being $l_{i,t-1}$ and f_{it}, respectively (that is $f_{it} + l_{i,t-1} = st_{it}$).

The second formulation is then as follows:

$$v(F2) = Min\left(\sum_{i=1}^{n}\sum_{t=1}^{m}(sc_{it}z_{it} + sc_{it}w_{it}) + \sum_{i=1}^{n}\sum_{t=1}^{m}\sum_{k=t}^{m} cs_{itk}x_{itk}\right) \tag{15}$$

Subject to:

$$\sum_{k=1}^{t} x_{ikt} = 1 \quad \forall i \in I, t \in T | d_{it} > 0 \tag{16}$$

$$\sum_{i=1}^{n} st_{it}z_{it} + \sum_{i=1}^{n}\sum_{k=t}^{m} vt_{it}d_{ik}x_{itk} + \sum_{i=1}^{n} l_{it} + \sum_{i=1}^{n} f_{it} \leqslant Cap_t \quad \forall t \in T \tag{17}$$

$$x_{itk} \leqslant z_{it} + w_{it} \quad \forall i \in I, t \in T, k \in T, k \geqslant t \tag{18}$$

$$f_{it} + l_{i,t-1} = w_{it} st_{it} \quad \forall i \in I, t \in T \tag{19}$$

$$\sum_{i=1}^{n} w_{it} \leqslant 1 \quad \forall t \in T \tag{20}$$

$$y_{it} \in \{0,1\}, w_{it} \in \{0,1\}, l_{it} \geqslant 0, l_{i0} = 0, f_{it} \geqslant 0 \quad \forall i \in I, t \in T \tag{21}$$

$$x_{itk} \geqslant 0 \quad \forall i \in I, t \in T, k \in T, k \geqslant t \tag{22}$$

The objective function (15) minimizes the total setup, production and inventory costs. The constraints (16) guarantee that the demand is satisfied in each period. Constraints (17) ensure that the total capacity consumed during a period for production and setups is less than or equal to the available capacity. The setup constraints (18) do not allow any production in period t unless a setup is done (either complete or crossover). When a setup is split, constraints (19) ensure that the split times add up to the total setup time. Constraints (20) state that the setup can be split across periods for at most one item. Finally, Constraints (21) and (22) define the variable domains.

3.3. Omitting the binary conditions on the crossover variables in F1

Analyzing the constraints (10)–(12) we observe that the integrality constraints on v_{it} can be dropped in $F1$. A setup for product i is defined as 'active' in period t if the corresponding setup variable (i.e. $y_{i,t}$) equals 1. The idea is that it is always feasible to limit the allowable time for a setup crossover to the maximum of the active setup times in a period (as formally discussed in Proposition 1). If less time is allowed (because there is not enough idle capacity in the previous period) or needed, the u_t variables can always assume a lower value. This constraint is still imposed if we drop the integrality constraints on the v_{it} variables (assuming positive setup times). The right-hand side of (10) cannot be more than the maximum of the active setup times because of constraints (11) and (12) together even if the binary conditions on the v_{it} variables are dropped. Note that when the binary conditions are dropped, the variable v_{it} does not necessarily indicate anymore which item is split, since it can assume fractional values. The variable v_{it} can take any fractional value between 0 and 1, and indicates the faction of each setup time utilized to determine the maximum allowed time to be borrowed for the setup crossover. The formulation $F3$ consists of the objective function (6), subject to constraints (7)–(14) with the integrality constraints on v_{it} dropped.

Note that $v(F)$ indicates the optimal objective function value of formulation F.

Proposition 1. $v(F3) = v(F1)$.

Proof. Observe that there is an incentive to make the right-hand side of (10) as large as possible, in order to allow the maximum flexibility. The v_{it} variable does not appear in the objective function, and the values are constrained by inequalities (11), (12) and the domain restrictions. Therefore, there exists an optimal solution for $F1$ in which the right-hand side of (10) $\sum_{i=1}^{n} v_{i,t-1}st_{it} = \max_{i \in I|y_{it}=1}\{st_{it}\}$. When we drop the integrality constraints on the v_{it} variables, the right-hand side of (10) will still have $\max_{i \in I|y_{it}=1}\{st_{it}\}$ as the maximum value. Therefore, by dropping the integrality constraints we will obtain the same objective function value as with the integrality constraints. □

3.4. Symmetry Breaking Constraints

As we will see in this section, it is possible that the two formulation proposed in the literature (($F1$) and ($F2$)) have alternative (optimal) solutions with the same (optimal) objective function value. The problem with these alternative or symmetric solutions is that they can increase the total computation time needed due to duplication in the branch-and-bound tree (see e.g. Sherali & Smith, 2001; Jans, 2009; Jans & Desrosiers, 2013).

Fig. 1 shows a small example where symmetric solutions exist for formulations ($F1$) and ($F2$). In Fig. 1, white blocks represent pro-

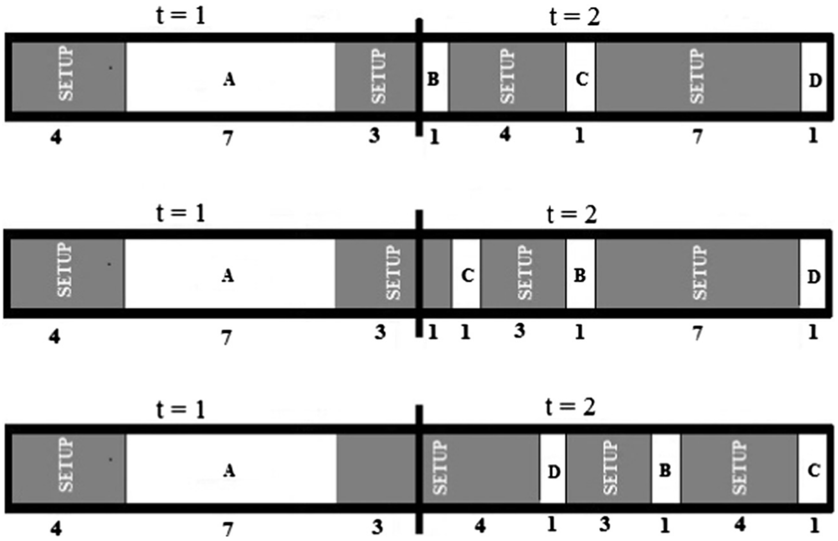


Fig. 1. Graphical solution of an example with symmetric solutions.

duction time that is consumed in that period, dark gray represents the setup time. Observe that we have three alternative feasible solutions producing the same items, consuming the same amount of capacity and with the same total cost. In all these solutions, there is a setup crossover between periods 1 and 2, but for different products.

3.4.1. Model to break the symmetry of formulation F1 ($F1'$)

The next proposition formally states that for formulation $F1$, it is possible that alternative (optimal) solutions exist with the same (optimal) objective function value.

Proposition 2. *Given a feasible (or optimal) solution for* $F1$*, with a setup crossover for product i from period t to* $t+1$ *(i.e.* $v_{it} = 1$ *and* $v_{jt} = 0 \ \forall j \in I \setminus \{i\}$*), we can construct an alternative feasible (or optimal) solution with the same objective function value if there exists in period* $t+1$ *an active setup for another product* i' *which has an equal or higher setup time (i.e.* $st_{i',t+1} \geqslant st_{i,t+1}$ *and* $y_{i',t+1} = 1$*). This solution can be constructed as follows:*

$v_{i't} = 1, v_{it} = 0$ *and all other variables (including* u_t*) remaining the same.*

Proof. The proof is easily established by the following two reasons:

(1) The new solution satisfies all the constraints.
(2) We have the same objective function, because the values of the variables x_{itk} and y_{it} remain the same. □

We can exclude these alternative solutions by explicitly imposing that the item with the highest active setup time in period $t+1$ is always chosen as the item for which we have a setup crossover between periods t and $t+1$. This is always feasible since the variable u_t can take a value which is lower than this setup time, or can even take the value of zero (if no idle capacity is available in period t, or if a setup crossover is not beneficial because there is ample capacity in period $t+1$). To impose this condition, we first have to order all the items in a decreasing order of their setup times. Next we have to add the following symmetry breaking constraints to formulation $F1$:

$$v_{1,t-1} = y_{1t} \quad \forall t \in T \setminus \{1\} \tag{23}$$

$$v_{i,t-1} \geqslant y_{it} - \sum_{j=1}^{i-1} y_{jt} \quad \forall i \in I \setminus \{1\}, \forall t \in T \setminus \{1\} \tag{24}$$

Remember that v_{it} is a binary variable, indicating if the setup is split between period t and period $t+1$ for item i. Constraints (23) and (24) impose in each period the setup crossover for the product with the highest active setup time. Note that as the items are ordered according to the decreasing order of setup time, item 1 has the highest setup time. Therefore, constraint (23) enforces the setup crossover between periods $t-1$ and t for the first item (i.e. the one with the highest setup time) only if this item is setup in period t. Constraint (24) enforces a setup crossover between periods $t-1$ and t for item i only if this item is setup in period t and if none of the items with a higher setup time have been setup in period t. Note that constraint (11) still prevents a crossover for an item if there is no setup. Formulation $F1$ augmented with constraints (23) and (24) will be called $F1'$.

3.4.2. Model to break the symmetry of formulation F2 ($F2'$)

For formulation $F2$, we observe as well that there can be several alternative solutions with the same objective function value. The reason is basically the same as for formulation $F1$.

Proposition 3. *Given a feasible (or optimal) solution for* $F2$*, with a setup crossover for product i from period t to* $t+1$ *(i.e.* $w_{i,t+1} = 1, w_{j,t+1} = 0 \ \forall j \in I \setminus \{i\}$ *and* $z_{i,t+1} = 0$*), we can construct an alternative feasible (or optimal) solution with the same objective function value if there exists in period* $t+1$ *an active setup for another product* i' *which has an equal or higher setup time (i.e.* $st_{i',t+1} \geqslant st_{i,t+1}$ *and* $z_{i',t+1} = 1$*). This solution can be constructed as follows:*

$$w_{i',t+1} = 1, z_{i',t+1} = 0$$

$$w_{i,t+1} = 0, z_{i,t+1} = 1$$

$$l_{i't} \longleftarrow l_{it}, l_{it} = 0$$

$$f_{i',t+1} = st_{i',t+1} - l_{i't}, f_{i,t+1} = 0$$

Proof. The proof is established by the following two reasons:

(1) The new solution remains feasible. Indeed:
(16) is satisfied for items i and i' since the x_{ikt} variables do not change.
The left-hand side of (17) for period t remains unchanged since $l_{i't}$ has taken the value of l_{it} and l_{it} has taken the value of zero, so that $\sum_{j=1}^{n} l_{jt}$ remains the same in the two solutions.
The left-hand side of (17) for period $t+1$ also has the same value after the changes (see Table 1).
(18) is satisfied since the right-hand side for item i and i' is still equal to 1.
(19) is satisfied for items i and i' by construction;
(20) is satisfied since we still only have one item (in each period) for which we allow a setup crossover.
(2) We have the same objective function by construction. □

We also propose a symmetry breaking constraint for formulation $F2$ to impose the setup crossover always for the product with the highest active setup time. As in the previous formulation, we first have to order all the items according to a decreasing order of setup times and then we add to formulation $F2$ the following new constraints:

$$\sum_{j=1}^{i-1} w_{jt} \geqslant z_{it} \quad \forall t \in T, i \in I \setminus \{1\} \tag{25}$$

Constraints (25) impose that if there is a complete setup for item i, this means there must have been a partial setup for an item $j < i$ (assuming an decreasing order of setup time). Constraints (25) will hence assign the setup crossover to the item with the highest active setup time. Note that this partial setup for item j might be a case where there is no time borrowed from the previous period (i.e. $w_{jt} = 1$ and $l_{j,t-1} = 0$ and $f_{jt} = st_{it}$). Formulation $F2$ augmented with constraints (25) will be called $F2'$.

3.5. Example

The following example shows the solutions for all formulations applied to the same instance. We adapted the example proposed in Belo-Filho et al. (2014) making some changes considering that in their case setup carryover is allowed.

We have to determine a production plan for four different items $i = \{A, B, C, D\}$ over a planning horizon composed of five periods. Tables 2 and 3 contain the parameters, the demand and capacity values. Note that the parameters in Table 2 are time independent.

All formulations were solved to optimality using this data set. Fig. 2 and Table 4 illustrate the graphical solutions and the relevant non-zero variable values, respectively.

Additionally to Fig. 1, in Fig. 2 light gray represents idle time. The values of the non-zero decision variables can be found in Table 4. For the classical formulation ($F0$) the value of the optimal solution is 688. This high value results mainly from the inventory for item B from period 1 to period 3 (20 units), for item C from period 2 to period 3 (30 units) and for item D from period 3 to period 5 (40 units). Note that in this example the inventory costs are very high. However, due to the lack of capacity it is impossible to have a setup for each item in a period with positive demand, and the optimal solution for $F0$ results in high inventory levels.

Table 1
Left hand side of constraint (17) for period $t+1$ and items i and i'.

	Old solution	New solution
$\sum_{i=1}^{n} st_{i,t+1} z_{i,t+1}$	$st_{i',t+1} \times 1$	$st_{i,t+1} \times 1$
$\sum_{i=1}^{n} l_{i,t+1}$	Unchanged	Unchanged
$\sum_{i=1}^{n} f_{i,t+1}$	$f_{i,t+1} = st_{i,t+1} - l_{it}$	$f_{i',t+1} = st_{i',t+1} - l_{i't}$
Total	$st_{i',t+1} + st_{i,t+1} - l_{it}$	$st_{i,t+1} + st_{i',t+1} - l_{i't}$

Although slightly different, the solutions found by the formulations $F1$, $F1'$, $F2$, $F2'$ and $F3$ have the same objective function value of 22. The only difference is that in formulation $F1$ and $F1'$ the setup for item A is split between periods 1 and 2 and for the formulations $F2$, $F2'$ and $F3$ the setup for this item is completely done in period 2. Observe that the solution obtained by $F2, F2'$ and $F3$ can directly be transformed into the solution obtained by $F1$ and $F1'$ by splitting the setup of product A over periods 1 and 2. We see hence that there can be equivalent alternative optimal solutions. Note that there is no inventory in the solutions obtained by these formulations. Note also that although formulations $F2$ and $F2'$ present the same solution, the value of some variables are different (see Table 4). This occurs because for a product with a partial setup between periods $t-1$ and t ($w_{it} = 1$), it is still possible in some cases to do the complete setup in period t by choosing $l_{i,t-1} = 0$ or we do the complete setup in period $t-1$ by choosing $f_{it} = 0$.

4. Computational results

The formulations were modeled in AMPL using CPLEX 12.6 as solver. The tests were done on a personal computer Intel Core-I5, 2.27 GHz with 6 GB of RAM and the Windows operating system. The computational tests involve five experiments based on standard instances proposed in Trigeiro et al. (1989). In the first experiment, the formulations are tested for the well known F and G instances. In the second experiment, the formulations were tested on a large data set of 540 standard instances. In the third experiment we test some adapted instances with high values for the inventory costs, while in the fourth one we test some instances considering the possibility of backlog. Finally, in the fifth experiment we test the instances from the first and second experiment disregarding the setup costs. We have limited the computational time in all experiments to 1800 s per instance. Note that in these instances the unit production costs are not considered.

4.1. Results for Experiment 1

The formulations were tested for a set of 145 instances proposed in Trigeiro et al. (1989). These are 70 instances from the F-set and 75 from the G-set. The F-set contains 70 instances with 6 items and 15 periods covering different levels of setup cost (low [520, 680], medium [200, 1000] and high [400, 2000]) and setup time (low [10, 50] and high [30, 150]). The G-set consists of 50 instances with 6 items and 15 periods and 5 instances for each of the cases with 12 items and 15 periods, 24 items and 15 periods, 6 items and 30 periods, 12 items and 30 periods and 24 items and 30 periods. Moreover, for the G-set the setup costs and times are always generated between [200, 1000] and [10, 50], respectively.

In Table 5 we give the upper bounds (Columns *UB*) and the computational times in seconds (Columns *Time*) for all formulations. We set the upper bounds found by the classical formulation ($F0$) to 100% and calculate the others values relative to this. As expected, all formulations with setup crossover have found better solutions compared to the classical formulation ($F0$) and the differences are bigger for problems with 6 items. Comparing the computational times, we can see that all formulations except formulations $F2$ and $F2'$ are faster than the classical formulation, whereas $F2$ is always slower and $F2'$ is faster only for instances with 30 periods.

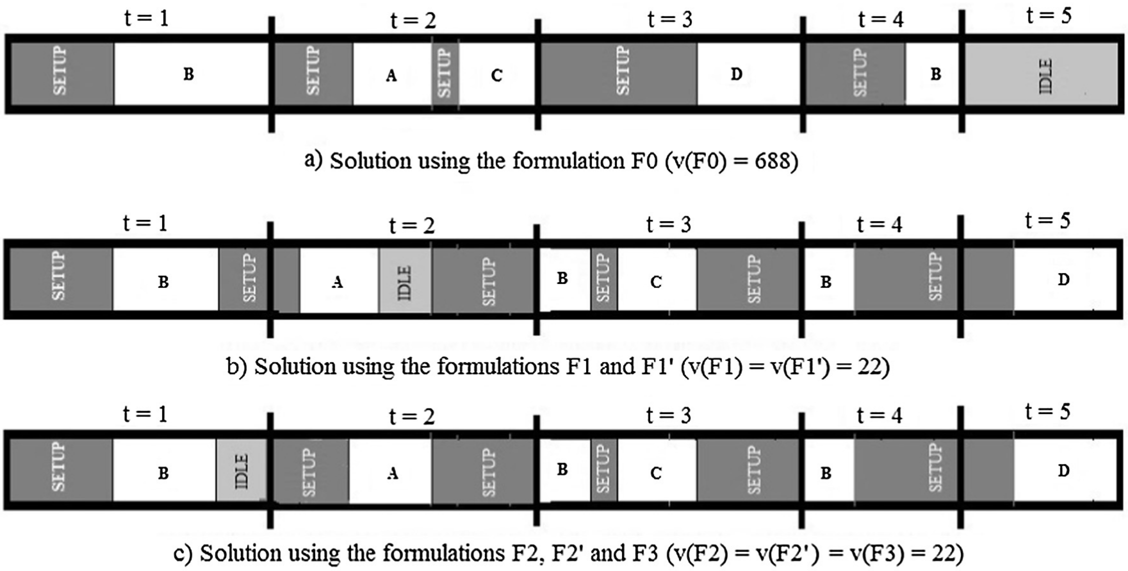
We observe that $F1$ is much faster than $F0$, which is surprising since it contains more binary variables. Omitting the binary condi-

Table 2
Parameters for the example.

	A	B	C	D
vt_i	0.1	0.1	0.1	0.1
hc_i	3	4	1	6
st_i	3	4	1	6
sc_i	3	4	1	6

Table 3
Demand and capacity data.

d_{it}	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
$i = A$	0	30	0	0	0
$i = B$	40	0	20	20	0
$i = C$	0	0	30	0	0
$i = D$	0	0	0	0	40
Cap_t	10	10	10	6	6



a) Solution using the formulation F0 (v(F0) = 688)

b) Solution using the formulations F1 and F1' (v(F1) = v(F1') = 22)

c) Solution using the formulations F2, F2' and F3 (v(F2) = v(F2') = v(F3) = 22)

Fig. 2. Graphical solution of the example for all formulations.

tion on the v_{it} variables as done in formulation $F3$ does not result in a significant change in the CPU time. It provides a very small decrease compared to $F1$ for the data sets F, $G6-15$, $G6-30$ and $G24-30$ and a very small increase for the data sets $G12-15$, $G24-15$ and $G12-30$.

Note that there is no benefit in adding symmetry breaking constraints to formulation $F1$ considering that the CPU times of the formulation $F1'$ are bigger than $F1$. However, for formulation $F2$ the symmetry breaking constraints are very efficient and the difference in CPU times between the formulations $F2$ and $F2'$ is very significant. CPU times for $F2$ are at least more than 5 times higher than for $F2'$.

Note also that these instances are quite easy, considering that CPLEX has solved relatively fast almost all instances for all formulations except formulation $F2$. Moreover, with formulations $F1, F1', F2'$ and $F3$ the solver has proven the optimality for all instances within the time limit. Using formulations $F0$ and $F2$ the solver has proven the optimality for 98.6% and 88.9%, respectively.

4.2. Results for Experiment 2

In this experiment, the formulations were tested on a total of 540 instances with 20 periods, such that five characteristics are analyzed: number of items (10, 20 and 30), demand variability (medium $[0,125]$ and high $[0,200]$), setup cost (low $[25,75]$, medium $[100,300]$ and high $[400,1200]$), setup time (low $[5,17]$ and high $[21,65]$) and capacity utilization (low $[75\%]$, medium $[85\%]$, and high $[95\%]$). The numbers in the brackets indicate a uniform distribution between the two numbers. For more details on the data set, we refer to Trigeiro et al. (1989).

Table 6 shows the overall performance of the formulations. We report the relative upper bounds (UB), computational times in seconds ($Time$) and percentage of instances solved to optimality within the limit of 1800 s (OS). Since the symmetry breaking constraints in $F1'$ were not able to improve the results obtained by $F1$, the results of $F1'$ are omitted.

The overall analysis of Table 6 confirms the tendencies observed in Table 5. $F0$ is slower than $F1$ and $F3$. $F2$ is overall the slowest formulation and the performance of $F2'$ is significantly better than $F2$. We see that $F1$ and $F3$ generally provide a similar performance (both in terms of CPU times and the percentage of optimal solutions found). We also observe that the cost decrease obtained by introducing the possibility of a setup crossover is very small in these instances and that the relevance of including a setup crossover is bigger for problems with few items. The average cost

Table 4
Variables values of example for all formulations.

			F0 variables		
$x_{A22} = 1$	$x_{B11} = 1$	$x_{B13} = 1$	$x_{B44} = 1$	$x_{C23} = 1$	$x_{D35} = 1$
$y_{A2} = 1$	$y_{B1} = 1$	$y_{B4} = 1$	$y_{C2} = 1$	$y_{D3} = 1$	
			F1 and F1′ variables		
$x_{A22} = 1$	$x_{B11} = 1$	$x_{B33} = 1$	$x_{B44} = 1$	$x_{C33} = 1$	$x_{D55} = 1$
$y_{A2} = 1$	$y_{B1} = 1$	$y_{B3} = 1$	$y_{B4} = 1$	$y_{C3} = 1$	$y_{D5} = 1$
$u_1 = 2$	$u_2 = 4$	$u_3 = 4$	$u_4 = 4$		
$v_{A1} = 1$	$v_{B2} = 1$	$v_{B3} = 1$	$v_{D4} = 1$		
			F2 variables		
$x_{A22} = 1$	$x_{B11} = 1$	$x_{B33} = 1$	$x_{B44} = 1$	$x_{C33} = 1$	$x_{D55} = 1$
$z_{A2} = 1$	$z_{B1} = 1$	$z_{C3} = 1$			
$w_{B3} = 1$	$w_{B4} = 1$	$w_{D5} = 1$			
$l_{B2} = 4$	$l_{B3} = 4$	$l_{D4} = 4$			
$f_{D5} = 2$					
			F2′ variables		
$x_{A22} = 1$	$x_{B11} = 1$	$x_{B33} = 1$	$x_{B44} = 1$	$x_{C33} = 1$	$x_{D55} = 1$
$z_{C3} = 1$					
$w_{A2} = 1$	$w_{B1} = 1$	$w_{B3} = 1$	$w_{B4} = 1$	$w_{D5} = 1$	
$l_{B2} = 4$	$l_{B3} = 4$	$l_{D4} = 4$			
$f_{A2} = 3$	$f_{B1} = 4$	$f_{D5} = 2$			
			F3 variables		
$x_{A22} = 1$	$x_{B11} = 1$	$x_{B33} = 1$	$x_{B44} = 1$	$x_{C33} = 1$	$x_{D55} = 1$
$z_{A2} = 1$	$y_{B1} = 1$	$y_{B3} = 1$	$y_{B4} = 1$	$y_{C3} = 1$	$y_{D5} = 1$
$u_2 = 4$	$u_3 = 4$	$u_4 = 4$			
$v_{A1} = 1$	$v_{B2} = 1$	$v_{B3} = 1$	$v_{D4} = 0.66$		

Table 5
Average general results for F and G data sets.

Model	F		G6-15		G12-15		G24-15		G6-30		G12-30		G24-30	
	UB	Time	UB	Time	UB	Time	UB	Time	UB	Time	UB	Time	UB	Time
F0	100	2.3	100	21.7	100	5.8	100	10.2	100	364.3	100	642.6	100	472.9
F1	99.36	1.1	99.03	3.4	99.86	3.3	99.85	7.3	99.38	17.3	99.72	82.3	99.97	89.7
F1′	99.36	1.3	99.03	3.5	99.86	4.3	99.85	11.6	99.38	24.7	99.72	112.0	99.97	131.4
F2	99.36	87.0	99.03	102.9	99.86	401.1	99.85	569.1	99.38	936.4	99.72	1444.4	99.97	1332.5
F2′	99.36	7.7	99.03	21.5	99.86	16.0	99.85	17.7	99.38	136.0	99.72	211.3	99.97	228.7
F3	99.36	0.9	99.03	3.1	99.86	3.6	99.85	7.5	99.38	16.5	99.72	82.4	99.97	83.3

Table 6
General average results aggregated per number of items.

Model	10 items			20 items			30 items			Aver. OS	
	UB	Time	OS	UB	Time	OS	UB	Time	OS	UB	Time
F0	100	578.1	71.6	100	666.0	65.0	100	688.3	63.9	100	9.5
F1	99.40	452.6	80.0	99.78	605.4	68.9	99.86	601.1	69.4	99.85	3.8
F2	99.43	791.5	58.3	99.79	849.8	55.0	99.89	846.4	56.1	99.85	86.9
F2′	99.42	612.8	69.4	99.79	665.8	65.5	99.89	660.2	66.1	99.85	8.4
F3	99.40	467.5	78.3	99.77	590.3	70.0	99.83	598.3	68.9	99.85	4.3

decrease is 0.60% for 10 items, 0.23% for 20 items and 0.17% for 30 items. The average cost decrease over all 540 instances is 0.33%.

Table 6 shows that although the formulations $F1, F2, F2'$ and $F3$ have the same optimal solutions, $F3$ found slightly better solutions for problems with 20 and 30 items within the given time limit. Considering only instances for which CPLEX proved optimality (columns Aver. OS) we clearly see the big improvement obtained by including the symmetry breaking constraints in the formulation $F2$. The CPU times for formulations $F2$ and $F2'$ are 86.9 and 8.4, respectively.

Aiming to do a further analysis of the effect of introducing a setup crossover, Table 7 shows the characteristics of the solutions for 10 items for the formulations $F0$ and $F1$: the percentage of setup and holding cost (columns $SC(\%)$ and $HC(\%)$) in the objective function value and the number of setups and total inventory (columns setup and inv.). We observe that overall the number of setups is very similar for both formulations and the main difference is the level of inventory. We obtain a decrease in total inventory of approximately 1%, but the total inventory holding costs constitute only 30% of the total cost. So the overall cost decrease is relatively small. It explains the small decrease obtained by introducing the possibility of a setup crossover in these instances given that the value of the inventory costs are very low. Even though globally the total number of setups does not significantly change when we introduce a setup crossover, we see that the setup cost and the capacity tightness have an impact. Tight capacity levels and low setup cost generally lead to a slight increase in the total number of setups when allowing a setup crossover. The level of the setup times and the demand variability do not have a large impact. The overall analysis of these instances indicates that the benefits of a setup crossover come mainly from the decreased inventory level which results from a better matching of demand and supply

Table 7
Detailed results for 10 items.

		Model F0					Model F1				
		UB	SC (%)	HC (%)	Setup	Inv.	UB	SC (%)	HC (%)	Setup	Inv.
Capacity	Loose	100	74.50	25.50	100.6	11242	99.90	74.52	25.48	100.4	11,189
	Normal	100	73.79	26.21	102.5	10,995	99.62	73.83	26.17	102.2	11,010
	Tight	100	61.49	38.51	92.6	12,953	98.67	62.60	37.40	93.0	12,614
S. cost	Low	100	82.21	17.79	152.0	1878	99.11	83.08	16.92	152.5	1766
	Normal	100	65.50	34.50	90.4	9186	99.56	65.74	34.26	90.3	9088
	High	100	62.07	37.93	53.2	24,126	99.51	62.13	37.87	52.9	23,959
S. time	Low	100	69.28	30.72	98.7	11,521	99.62	69.52	30.48	98.6	11,427
	High	100	70.57	29.43	98.3	11,939	99.17	71.12	28.88	98.5	11,782
Demand	Medium	100	71.15	28.85	105.3	12,112	99.48	71.41	28.59	105.3	12,053
	High	100	68.71	31.29	91.7	11,348	99.31	69.23	30.77	91.8	11,155
Average		100	69.93	30.07	98.5	11,730	99.40	70.32	29.68	98.5	11,604

through the increased flexibility. This might require, however, a slight change in the number of setups as well.

It is important to note that although the cost decrease obtained by introducing the possibility of a setup crossover is on average small in these instances (i.e. 0.33%), for some cases this decrease is more relevant. A more detailed analysis revealed that the cost decrease obtained by introducing the possibility of a setup crossover is the biggest for configurations with 10 items, tight capacity, low setup cost, high setup time and medium and high demand variability, where we obtained respectively a 2.77% and 2.62% cost decrease by including a setup crossover. On the other hand, for many configurations with loose capacity we did not obtain any improvement. This is in line with the analyses presented in Tables 6 and 7.

4.3. Results for Experiment 3

In this experiment the formulations were tested on a set of 180 instances. These are the same as the instances with 10 items of experiment 2 with an altered high value for the inventory costs. To generate the instances with high inventory costs we multiply the inventory costs by 10 and 100.

Tables 8 and 9 show the benefits of considering setup crossover for problems where the inventory costs are significant. The global analysis confirms some of the conclusions of the previous experiments. $F1$ and $F3$ have a similar performance. $F2$ is again the slowest formulation. However, in contrast to the results of the previous experiments, we do see a significant decrease in the total costs when setup crossover is allowed, which is on average 3% (for the inventory cost × 10) and 7.8% (for the inventory cost × 100).

Table 8 contains the results for the instances with the inventory costs multiplied by 10. We observe that the benefits of a setup crossover are the highest in a setting with tight capacity (4.5% cost decrease), high setup times (4.6% cost decrease) and low setup cost (4.7% cost decrease).

Table 9 contains the results for instances with inventory costs multiplied by 100. We observe that for these instances, the effect of allowing a setup crossover is the highest for the instances with normal capacity, low setup cost and high setup time where the total cost decreases more than 10% on average. Note also that even in a setting with loose capacity the total cost decrease is almost 7%.

Tables 10 and 11 show the behavior of the solutions for the results with inventory costs multiplied by 10 and 100, respectively. We observe that, contrary to the results of Table 7, the percentage of inventory cost in the objective function value is very high especially for instances in which the inventory costs are multiplied by 100 (63.23% for formulation $F0$ and 60.24% for formulation $F1$). It explains the more significant decrease obtained by introducing the possibility of a setup crossover in these instances given that the value of the inventory costs are very high. We observe that for the instances in which the inventory costs are multiplied by 10 (Table 10) the total inventory goes down by approximately 5.5%, and the total setups only increase by 0.5%. Regarding the instances in which the inventory costs are multiplied by 100 (Table 11) the total inventory goes down by approximately 6.5% and the total setups only increase by 0.5%.

4.4. Results for Experiment 4

In this experiment the formulations were adapted to allow backlog (Pochet & Wolsey, 2006) and were tested on a set of 60

Table 8
Average general results with inventory costs multiplied by 10.

		Model F0		Model F1		Model F2		Model F2'		Model F3	
		UB	T (s)	UB	T (s)	UB	T (s)	UB	T (s)	UB	T (s)
Capacity	Loose	100	84	98.95	2	98.95	155	99.95	36	98.95	2
	Normal	100	437	96.55	280	96.56	836	96.55	489	96.54	270
	Tight	100	1345	95.48	1182	95.51	1744	95.48	1488	95.47	1170
S. cost	Low	100	260	95.33	194	95.33	544	95.33	376	95.33	193
	Normal	100	697	97.07	454	97.08	977	97.05	727	97.05	442
	High	100	910	98.57	815	98.60	1214	98.60	911	98.57	805
S. time	Low	100	456	98.59	428	98.61	802	98.60	623	98.60	421
	High	100	789	95.39	548	95.40	1021	95.38	719	95.37	540
Demand	Medium	100	834	97.49	687	97.51	984	97.49	892	97.48	686
	High	100	410	96.49	289	96.50	840	96.49	450	96.49	275
Average		100	622	96.99	488	97.00	912	96.99	671	96.99	481

Table 9
Average general results with inventory costs multiplied by 100.

		Model F0		Model F1		Model F2		Model F2’		Model F3	
		UB	T (s)	UB	T (s)	UB	T (s)	UB	T (s)	UB	T (s)
Capacity	Loose	100	153	93.02	34	93.03	186	93.02	118	93.02	34
	Normal	100	450	89.51	271	89.54	839	89.51	471	89.51	250
	Tight	100	1199	94.05	1046	94.09	1742	99.07	1419	94.04	1022
S. cost	Low	100	258	89.60	191	89.61	543	89.60	350	89.60	183
	Normal	100	562	91.79	369	91.82	976	91.80	680	91.79	353
	High	100	983	95.19	790	95.23	1248	95.20	979	95.18	769
S. time	Low	100	358	95.86	349	95.88	794	95.87	555	95.86	343
	High	100	843	88.53	552	88.55	1051	88.53	784	88.52	528
Demand	Medium	100	825	93.11	677	93.14	1008	93.12	927	93.10	655
	High	100	376	91.28	223	91.29	837	91.28	412	91.28	216
Average		100	601	92.19	450	92.22	922	92.20	669	92.19	435

Table 10
General detailed results with inventory costs multiplied by 10.

		Model F0				Model F1			
		SC (%)	HC (%)	Setup	Inv.	SC (%)	HC (%)	Setup	Inv.
Capacity	Loose	88.21	11.79	158.3	1351	89.65	10.35	159.1	1228
	Normal	67.17	32.83	146.8	2868	69.71	30.29	147.6	2663
	Tight	22.62	77.38	108.9	9613	23.90	76.10	109.5	9216
S. cost	Low	63.79	36.21	164.1	1540	66.12	33.88	164.8	1404
	Normal	57.95	42.05	139.6	4037	59.84	40.16	140.5	3790
	High	56.26	43.74	110.4	8255	57.30	42.70	110.9	7914
S. time	Low	62.39	37.61	141.8	4039	63.16	36.84	142.1	3932
	High	56.28	43.72	134.2	5182	59.02	40.98	135.4	4807
Demand	Medium	63.49	36.51	145.6	4434	64.96	35.04	146.5	4189
	High	55.18	44.82	130.4	4787	57.21	42.79	131.0	4549
Average		59.33	40.66	138.0	4611	61.09	38.91	138.7	4369

Table 11
General detailed results with inventory costs multiplied by 100.

		Model F0				Model F1			
		SC (%)	HC (%)	Setup	Inv.	SC (%)	HC (%)	Setup	Inv.
Capacity	Loose	74.12	25.88	170.4	721	78.65	21.35	171.0	598
	Normal	32.90	67.10	153.5	2592	37.10	62.90	154.6	2358
	Tight	3.31	96.69	112.7	9668	3.56	96.44	113.2	9241
S. cost	Low	46.11	53.89	165.4	1539	50.02	49.98	166.3	1402
	Normal	33.02	66.98	144.3	4027	36.32	63.68	144.8	3767
	High	31.19	68.81	126.9	7415	32.95	67.05	127.7	7028
S. time	Low	44.22	55.78	151.5	3662	46.17	53.83	151.9	3542
	High	29.33	70.67	139.5	4992	33.36	66.64	140.6	4589
Demand	Medium	45.81	54.19	151.3	4108	48.96	51.04	151.8	3859
	High	27.74	72.26	139.7	4546	30.57	69.43	140.7	4273
Average		36.77	63.23	145.5	4327	39.76	60.24	146.2	4066

instances. These are the same as the instances with 10 items and tight capacity of experiment 2 with an altered (reduced) value for the capacity in order to generate some backlog. To generate these instances with very tight capacity we reduce the capacity by 5% and 10%. We set the backlog costs for each item equal to 100× inventory holding cost.

Tables 12–15 present the overall performance of the formulations for problems that consider the possibility of backlog (based on instances in which the two formulations found feasible solutions). We report all factors that have been analyzed and added the percentage of backlog cost (columns $B(\%)$) in the total objective function value, the percentage of feasible solutions (columns FS) and the number of total backlog (columns Back.).

The overall analysis of Tables 12–15 shows that for problems that allow the possibility of backlog there is a significant decrease in the total costs when a setup crossover is allowed, which is on average 2.3% and 4% for instances for which the capacity is reduced by 5% and 10%, respectively. We also observe, especially for instances for which the capacity is reduced by 10%, an increase in the number of feasible solutions (4%).

Tables 12 and 13 present the results for instances for which the capacity is reduced by 5%. We observe that for these instances when the setup cost and time is high, the importance of including a setup crossover increases. Note also that overall the number of setups is very similar again for the case with and without setup crossover. We obtain a decrease in total inventory and backlog of

Table 12
General results with backlog and capacity reduced by 5%.

		Model F0			Model F1		
		UB	T (s)	FS	UB	T (s)	FS
S. cost	Low	100	1401	100	97.86	1101	100
	Normal	100	1800	100	98.55	1666	100
	High	100	1715	80	96.58	1731	85
S. time	Low	100	1742	86	99.08	1758	90
	High	100	1539	100	96.57	1245	100
Demand	Medium	100	1800	93	97.86	1617	97
	High	100	1467	93	97.62	1350	93
Average		100	1633	93	97.73	1483	95

Table 13
General detailed results with backlog capacity reduced by 5%.

		Model F0						Model F1					
		SC (%)	HC (%)	B (%)	Setup	Inv.	Back.	SC (%)	HC (%)	B (%)	Setup	Inv.	Back.
S. cost	L	45.52	54.48	0	114.8	7676	0	46.97	53.03	0	115.5	7397	0
	N	47.59	50.34	2.05	77.5	15,345	10	48.84	49.17	1.97	78.2	14,766	10
	H	43.25	21.02	35.72	60.3	21,167	793	44.71	21.48	33.80	60.3	20,921	759
S. time	L	40.98	51.43	7.57	83.5	14,911	252	41.55	50.92	7.51	83.7	14,639	252
	H	49.63	36.51	13.85	88.0	13,714	211	51.71	35.46	12.82	88.8	13,246	192
Demand	M	44.44	41.86	13.69	90.3	14,143	341	45.73	41.08	13.18	90.9	13,739	335
	H	46.79	45.18	8.18	81.6	14,397	120	48.25	44.19	7.54	81.9	14,047	106
Average		45.61	43.44	10.93	85.9	14,270	230	46.99	42.64	10.36	86.4	13,893	221

Table 14
General results with backlog capacity reduced by 10%.

		Model F0			Model F1		
		UB	T (s)	FS	UB	T (s)	FS
S. cost	Low	100	1731	100	97.63	1411	100
	Normal	100	1800	55	95.21	1688	55
	High	100	1800	25	91.23	1800	35
S. time	Low	100	1800	37	96.57	1800	37
	High	100	1745	83	95.75	1440	90
Demand	Medium	100	1800	66	94.92	1800	70
	High	100	1714	53	97.36	1237	57
Average		100	1762	59	96.00	1550	63

Table 15
General detailed results with backlog capacity reduced by 10%.

		Model F0						Model F1					
		SC (%)	HC (%)	B (%)	Setup	Inv.	Back.	SC (%)	HC (%)	B (%)	Setup	Inv.	Back.
S. cost	L	27.68	48.80	23.50	91.2	12,399	781	29.01	47.58	23.39	92.4	12,083	775
	N	35.58	50.34	14.06	65.3	19,995	1152	36.82	49.38	13.79	65.3	19,021	959
	H	3.08	3.86	93.04	33.2	28,080	7925	3.38	4.03	92.57	33.4	26,682	6977
S. time	L	9.10	39.31	51.58	66.9	19,588	2525	9.25	39.43	51.30	66.8	19,164	2324
	H	34.41	44.67	20.91	78.9	15,714	1606	36.02	43.25	20.72	80.0	14,940	1415
Demand	M	22.78	39.22	37.99	73.5	18,012	2625	23.68	38.61	37.69	73.9	17,205	2299
	H	31.55	47.80	20.64	77.4	15,505	964	33.03	46.43	20.52	78.6	15,012	935
Average		26.68	43.03	30.28	75.2	16,898	1886	27.84	42.08	30.06	76.0	16,231	1693

approximately 2.7% and 4%, respectively. Finally, the percentage of backlog in the total objective function value is relatively small (on average only 11%) and there is no backlog for instances with low setup cost.

Tables 14 and 15 show the results for instances in which the capacity is reduced by 10%. For these instances the percentage of backlog in the total objective function value is higher (approximately 30%) and we obtain a decrease in total backlog of 10.2%. We observe that for instances with high setup cost, the percentage of backlog in the objective function value is 93%. Moreover, we obtained a decrease in the total costs of 8.8% for these instances.

4.5. Results for Experiment 5

In this experiment, the formulations were tested on a total of 685 instances. These are the same as the 145 instances of the first

Table 16
Average general results for F and G data sets.

Model	F		G6-15		G12-15		G24-15		G6-30		G12-30		G24-30	
	UB	Time	UB	Time	UB	Time	UB	Time	UB	Time	UB	Time	UB	Time
F0	100	6.3	100	14.8	100	5.4	100	404.9	100	394.3	100	1440.6	100	1800
F1	75.37	1.4	79.26	4.7	84.10	3.7	91.98	51.8	68.99	368.6	85.76	1248.7	86.42	1800
F2	75.37	578.1	79.26	402.15	84.10	1085.7	91.98	1800	69.07	1800	86.40	1800	86.74	1800
F2'	75.37	31.8	79.26	78.5	84.10	64.9	91.98	301.6	69.07	1175.7	85.87	1513.5	86.51	1800
F3	75.37	1.3	79.26	3.9	84.10	3.6	91.98	46.6	68.99	368.3	85.73	1379.2	86.44	1800

Table 17
General average results aggregated per number of items.

Model	10 items			20 items			30 items			Aver. OS	
	UB	Time	OS	UB	Time	OS	UB	Time	OS	UB	Time
F0	100	575.6	72.7	100	898.7	56.6	100	958.5	47.7	100	2.1
F1	83.91	408.1	82.7	88.32	751.6	61.1	91.33	877.6	54.4	79.57	0.7
F2	83.93	927.0	49.4	88.35	1062.6	43.8	91.35	1084.7	40.0	79.57	55.8
F2′	83.92	660.9	67.2	88.33	868.9	53.8	91.33	955.5	48.8	79.57	4.3
F3	83.91	396.6	85.5	88.32	757.4	62.2	91.32	879.8	54.4	79.57	0.7

Table 18
Detailed results for 10 items.

		Model F0					Model F1				
		UB	SC (%)	HC (%)	Setup	Inv.	UB	SC (%)	HC (%)	Setup	Inv.
Capacity	Loose	100	0	100	175.2	592.8	79.53	0	100	177.0	469.2
	Normal	100	0	100	156.1	2594.2	78.35	0	100	156.7	2356.5
	Tight	100	0	100	116.0	9673.6	93.85	0	100	116.0	9271.6
S. cost	Low	100	0	100	168.1	1539.5	78.07	0	100	169.6	1403.8
	Normal	100	0	100	148.4	3899.0	82.06	0	100	148.7	3649.1
	High	100	0	100	130.9	7422.2	91.61	0	100	131.4	7044.6
S. time	Low	100	0	100	154.6	3677.4	88.20	0	100	155.1	3560.6
	High	100	0	100	143.7	4896.4	79.62	0	100	144.7	4504.4
Demand	Medium	100	0	100	154.5	4114.2	83.51	0	100	155.5	3865.3
	High	100	0	100	143.8	4459.5	84.31	0	100	144.3	4199.6
Average		100	0	100	149.1	4286.9	83.91	0	100	149.9	4032.5

experiment plus the 540 instances of the second experiment but all the setup costs are set to zero. Lot sizing instances with setup times and zero setup costs were considered in the paper by Süral, Denizel, and Van Wassenhove (2009).

The overall analysis of Tables 16–18 shows that the benefits of considering setup crossover for problems without setup cost are really significant. Indeed, the results show that from a practical perspective, it is a highly relevant setting in which the setup crossover is very beneficial. It can be explained because only inventory holding costs are present in the objective function, and the previous experiments have shown that the possibility to do a setup crossover can substantially reduce the inventory levels.

Table 16 contains the results for the 145 instances of the Experiment 1. We observe again that the benefits of a setup crossover are the highest in a setting with few items. For instances with 6 items and 30 periods the average cost decrease was 31%. Although, the benefits of a setup crossover decrease when increasing the number of items, the average total cost decrease was still more than 13% for instances with 24 items and 30 periods. Note that for the same instances considering the setup cost (Experiment 1) we obtained a cost decrease of less than 1% on average for all instances.

Table 17 confirms the tendencies observed in Table 16 and in the Experiment 2. We see that the cost decrease obtained by introducing the possibility of a setup crossover is smaller for problems with many items. Furthermore, we see that, contrary to the results from Experiment 2, the benefits of the possibility of a setup crossover are very significant. We also observe that the computational times increase substantially compared to Experiment 2 for the instances with 20 and 30 items.

Table 18 shows the behavior of the solutions for 10 items for the formulations $F0$ and $F1$. We observe that different from the results from Experiment 2, there is a slight increase in the number of setups when considering the possibility of a setup crossover. However, the main difference is still the level of inventory. Note that again different from Experiment 2, for instances with tight capacity, the inclusion of a setup crossover is less relevant than the instances with loose and normal capacity. It occurs because for several easy problems (with loose and normal capacity) the problem considering setup crossover found optimal solutions with zero value (solutions without inventory) or near of zero (solutions with a low level of inventory) so the improvements in these cases are very significant. Finally, we see that the benefits of a setup crossover is more relevant for the instances with low setup cost where the average cost decrease is 21.93%.

5. Conclusion

In this paper, the lot sizing problem with capacity constraints and setup crossover is studied. Two different formulations have been proposed in the literature. Yet, no direct comparison has yet been done to determine which formulations performs better in computational experiments. A set of experiments clearly revealed that the formulation proposed by Menezes et al. (2010) performs much better than the formulation proposed by Mohan et al. (2012). Furthermore, we establish that in the formulation proposed by Menezes et al. (2010), the binary conditions on the new variables for the setup crossover are not necessary and they can be modeled as continuous variables. The computational experiments indicated that this only had a very minor impact. We established that several alternative solutions may exist for any given solution. The presence of such symmetrical solution typically leads to increased solution times. Therefore, we propose symmetry breaking constraints to eliminate these alternative solutions. The computational tests indicate that the symmetry breaking constraints are very effective for the formulation proposed by Mohan et al. (2012), but not for the formulation proposed by Menezes et al. (2010).

The computational experiments also allowed us to establish the benefits that can be derived by having the flexibility to perform a setup crossover compared to the case where such a crossover is not allowed. The benefits of the setup crossover depend on the characteristic of the problem. Especially for problems with zero setup costs or with high inventory holding cost the cost savings can be very significant. Indeed, for some instances with few items and zero setup cost, the cost decrease obtained by introducing the possibility of a setup crossover is more than 30%. Furthermore, when the inventory costs are multiplied by 10 and 100, the cost decrease is on average 3% and 7.8%, respectively. Finally, we also conclude that the benefits obtained by allowing a setup crossover are also significant for problems which allow the possibility of backlog. We obtained a decrease in the total costs by 2.3% and 4% when the capacity was decreased by 5% and 10%, respectively. Moreover, the formulation with a setup crossover found more feasible solutions.

Acknowledgements

This work was supported by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) (Process numbers 2010/16727-9, 2013/00965-6 and 2014/01203-5), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and by the Natural Sciences and Engineering Research Council of Canada [371966-2009]. This support is gratefully acknowledged.

References

Belo-Filho, M. A. F., Almada-Lobo, B., & Toledo, F. M. B. (2014). Models for capacitated lot-sizing problem with backlogging. Setup carryover and crossover. *Journal of the Operational Research Society, 65*(11), 1735–1747.

Camargo, V. C. B., Toledo, F. M. B., & Almada-Lobo, B. (2012). Three time-based scale formulations for the two-stage lot sizing and scheduling in process industries. *Journal of the Operational Research Society, 63*, 1613–1630.

de Araujo, S. A., de Reyck, B., Degraeve, Z., Fragkos, I., & Jans, R. (2015). Period decompositions for the capacitated lot sizing problem with setup times. *INFORMS Journal on Computing, 27*, 431–448.

Eppen, G. B., & Martin, R. K. (1987). Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research, 6*, 832–848.

Fiorotto, D. J., A, S., & Araujo, de (2014). Reformulation and a lagrangian heuristic for lot sizing problem on parallel machines. *Annals of Operations Research, 217*, 213–231.

Fiorotto, D. J., Jans, R., & de Araujo, S. A. (2014). An analysis of formulations for the capacitated lot sizing problem with setup crossover. *CIRRELT, 57*, 1–28.

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

Jans, R. (2009). Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. *INFORMS Journal on Computing, 21*, 123–136.

Jans, R., & Degraeve, Z. (2007). Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research, 177*, 1855–1875.

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

Jans, R., & Desrosiers, J. (2013). Efficient symmetry breaking formulations for the job grouping problem. *Computers & Operations Research, 40*(4), 1132–1142.

Kopanos, G. M., Puigjaner, L., & Maravelias, C. T. (2011). Production planning and scheduling of parallel continuous processes with product families. *Industrial and Engineering Chemistry Research, 50*(3), 1369–1378.

Krarup, J., & Bilde, O. (1977). Plant location, set covering and economic lot size: An O (mn)-algorithm for structured problems. *Numerische Methoden bei Optimierungsaufgaben. Bang 3: Optimierung bei Graphentheoritischen Ganzzahligen Problemen*, 155–186.

Manne, A. S. (1958). Programming of economic lot sizes. *Management Science, 4*, 115–135.

Menezes, A. A., Clark, A., & Almada-Lobo, B. (2010). Capacitated lot-sizing and scheduling with sequence-dependent, period-overlapping and non-triangular setups. *Journal of Scheduling, 14*(2), 209–219.

Mohan, S., Gopalakrishnan, M., Marathe, R., & Rajan, A. (2012). A note on modelling the capacitated lot-sizing problem with set-up carryover and set-up splitting. *International Journal of Production Research, 50*(19), 5538–5543.

Pochet, Y., & Wolsey, L. A. (2006). *Production planning by mixed integer programming*. Springer.

Sherali, H. D., & Smith, J. C. (2001). Improving discrete model representations via symmetry considerations. *Management Science, 47*(10), 1396–1407.

Sox, S. R., & Gao, Y. (1999). The capacitated lot sizing with setup carry-over. *IIE Transactions, 31*(2), 173–181.

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

Suerie, C., & Stadtler, H. (2003). The capacitated lot-sizing problem with linked lot sizes. *Management Science, 49*(8), 1039–1054.

Sung, C., & Maravelias, C. T. (2008). A mixed-integer programming formulation for the general capacitated lot-sizing problem. *Computers & Chemical Engineering, 32*, 244–259.

Süral, H., Denizel, M., & Van Wassenhove, L. N. (2009). Lagrangean relaxation based heuristics for lot sizing with setup times. *European Journal of Operational Research, 194*(1), 51–63.

Trigeiro, W. W., Thomas, J., & McClain, J. O. (1989). Capacitated lot sizing with setup times. *Management Science, 35*, 353–366.

Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management Science, 5*, 89–96.