CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

An Analysis of Formulations for the Capacitated Lot Sizing Problem with Setup Crossover

Diego Jacinto Fiorotto
Raf Jans
Silvio Alexandre de Araujo

November 2014

CIRRELT-2014-57

An Analysis of Formulations for the Capacitated Lot Sizing Problem with Setup Crossover

Diego Jacinto Fiorotto[1,2], Raf Jans[1,*], Silvio Alexandre de Araujo[2]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Logistics and Operations Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[2] Departamento de Matemática Aplicada, Universidade Estadual Paulista, 15054-000 São José do Rio Preto, SP, Brazil

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 to avoid the necessity of defining new extra binary variables to model the setup crossover and 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 analyses which factors influence this value.

Keywords. Lot sizing, setup crossover, mathematical formulation, symmetry breaking.

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 2011/22647-0) and by the Natural Sciences and Engineering Research Council of Canada (grant 342182-09). This support is gratefully acknowledged.

---

* Corresponding author: Raf.Jans@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2014

1 Introduction

The research on lot sizing 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 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 (Trigeiro et al., 1989)the setup for the first product type produced in a period starts at the beginning of that period (see Figure 1.a). 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 (see Figure 1.b). 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 moving (partially) the setup of the first product to the previous period. In the big bucket models, the setup times are smaller or equal than the capacity.

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 the period $t+1$ producing the item i 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 et al., 2012; Belo-Filho et al., 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 propose new ideas to avoid the necessity of defining new extra binary variables to model the setup crossover; 2) to propose new constraints to break the symmetry which is present in the formulations from the literature; 3) to compare the two formulations proposed in the literature to determine which formulation is the best; 4) to analyse the impact of the proposed adaptations of these formulations (i.e. no binary variables and symmetry breaking constraints), and 5) to determine the value of the flexibility provided by the setup crossover and analyse the factors that have an impact on this value.

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. In Section 4, we present some theoretical results for the formulations. Section 5 describes the computational results and analyses and finally in Section 6, 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 (see e.g. Trigeiro et al. (1989)). These models have been extended to deal with various industrial issues (see Jans and 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 (Krarup and Bilde, 1977) and their computational tests have shown that this formulation is better than formulations proposed by Sox and Gao (1999). Gopalakrishnan et al. (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 produce more feasible and improved solutions.

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 et al. (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 et al. (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 and 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) and three different ways to model the setup crossover without defining new extra binary variables. Finally, we 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.

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 the strategy used (Fiorotto and 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. 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 on 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

$I = \{1, ..., n\}$: set of items;

$T = \{1, ..., m\}$: 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 one unit of item i in period t to satisfy 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 ($F0$)

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

Subject to:

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

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

$$x_{itk} \leq y_{it} \qquad \forall i \in I, t \in T, k \in T, k \geq t \tag{4}$$

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

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 variables 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 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 ways to model the setup crossover are similar to that of the papers previously mentioned, and therefore they will be omitted.

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

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 $t+1$.

The first mathematical formulation based on the ideas proposed by Menezes et al. (2010) is as follows:

- Model adding extra binary variables ($F1$)

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

Subject to:

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

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

$$x_{itk} \leq y_{it} \qquad \forall i \in I, t \in T, k \in T, k \geq t \tag{9}$$

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

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

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

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

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

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 done. 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.

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 others new variables:

New decision variables:

z_{it}: $= 1$ if a complete setup is done in period t for item i, 0 otherwise;

v_{it}: $= 1$ if setup crossover between period $t-1$ and period t for item i with splits 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:

- Model separating complete setups ($F2$)

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

Subject to:

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

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

$$x_{itk} \leq z_{it} + v_{it} \qquad \forall i \in I, t \in T, k \in T, k \geq t \tag{18}$$

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

$$\sum_{i=1}^{n} v_{it} \leq 1 \qquad \forall t \in T \tag{20}$$

$$y_{it} \in \{0,1\}, \;\; v_{it} \in \{0,1\}, \;\; l_{it} \geq 0, \;\; l_{i0} = 0, \;\; f_{it} \geq 0 \qquad \forall i \in I, t \in T \tag{21}$$

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

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 Proposed Models

The first new proposed formulation is built upon the idea that it is possible to limit the quantity of borrowed time (u_t) in each period by the lowest value of the setup times. Although we restrict the set of feasible solution, we avoid the necessity of defining extra binary variables for the setup crossover. Note that this formulation will have a higher (or equal) optimal objective function value compared to formulations $F1$ and $F2$ (as we will formally discuss in Section 4) and hence can be used as a heuristic. This new formulation is defined as follows:

- Model with minimum setup time ($F3$)

$$v(F3) = Min \sum_{i=1}^{n} \sum_{t=1}^{m} sc_{it} y_{it} + \sum_{i=1}^{n} \sum_{t=1}^{m} \sum_{k=t}^{m} cs_{itk} x_{itk} \tag{23}$$

Subject to:

$$\sum_{k=1}^{t} x_{ikt} = 1 \qquad \forall i \in I, t \in T \mid d_{it} > 0 \tag{24}$$

$$\sum_{i=1}^{n} st_{it} y_{it} + \sum_{i=1}^{n} \sum_{k=t}^{m} vt_{it} d_{ik} x_{itk} + u_t \leq Cap_t + u_{t-1} \qquad \forall t \in T \tag{25}$$

$$x_{itk} \leq y_{it} \qquad \forall i \in I, t \in T, k \in T, k \geq t \tag{26}$$

$$u_{t-1} \leq \min_{\forall j \in I} \{st_{jt}\} \qquad \forall t \in T \tag{27}$$

$$y_{it} \in \{0,1\},\ x_{itk} \geq 0,\ u_{t-1} \geq 0,\ u_0 = 0 \quad \forall i \in I, t \in T, k \in T, k \geq t \tag{28}$$

The objective function (23) and the constraints (24), (25) and (26) are the same as constraints (6), (7), (8) and (9) in the formulation $F1$. Constraints (27) limit the extra time allowed for a setup crossover to the lowest setup time of all products. The last constraints (28) state the domain of the variables.

- Model with minimum active setup time ($F4$)

The second proposed formulation is an extension of the first one. The idea is that we can only use extra capacity from period $t-1$ for a product that is setup in period t, i.e., we can limit the borrowed extra time (in period $t-1$) to the minimum time of the active setups in period t.

The following modification of the constraints (27) handles this extension:

$$u_{t-1} \leq st_{it} y_{it} + \max_{\forall j \in I} \{st_{jt}\}(1 - y_{it}) \qquad \forall i \in I, t \in T \tag{29}$$

Constraints (29) limit the extra time allowed for a setup crossover to the minimum setup time of the active setups. If the setup is not active in period t for product i, i.e. $y_{it} = 0$, then the extra time is limited by the maximum setup time. Otherwise, if the setup is active for product i, i.e. $y_{it} = 1$, then the extra time is limited by exactly this setup time. As we have this constraint for every item, we are limiting the u_{t-1} variables to the minimum active setup time.

The second proposed formulation, $F4$, is the same as $F3$ with constraints (27) replaced by the constraints (29). This formulation will also result in a higher (or equal) optimal objective function value when compared to formulations $F1$ and $F2$, and hence can only be used as a heuristic.

- Model dropping the extra binary variables ($F5$)

The third new formulation is based on formulation $F1$. Analyzing the constraints (10) to (12) we observe that the integrality constraints on v_{it} can be dropped. 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 Section 4). 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 decisions are dropped, the variables v_{it} does not necessarily indicate anymore which item is split, since it can assume fractional values. They are only used to determine the maximum time allowed for the crossover. The formulation $F5$ consists of the objective function (6), subject to constraints (7)-(14) with the integrality constraints on v_{it} dropped.

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

For formulation $F1$, it is possible that alternative (optimal) solutions exists with the same (optimal) objective function value, as will be formally explained in Section 4. 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 and Smith (2001), Jans (2009) and Jans and Desrosiers (2013)). 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). To impose this condition, we first have to order the items in a decreasing order of their setup times. Next we have to add the following symmetry breaking constraints to formulation $F1$:

$$v_{1,t-1} = y_{1t} \qquad \forall t \in T \setminus \{1\} \tag{30}$$

$$v_{i,t-1} \geq y_{it} - \sum_{j=1}^{i-1} y_{jt} \qquad \forall i \in I \setminus \{1\}, \forall t \in T \setminus \{1\} \tag{31}$$

Constraints (30) and (31) 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 one has the highest setup time. Therefore, constraint (30) 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 (31) 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 (30) and (31) will be called $F1'$.

- 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 (see Lemma 3 in Section 4). The reason is basically the same as for formulation $F1$.

We also have proposed a type of symmetry breaking constraint to 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 the items according to a decreasing order of setup times and then we add to formulation $F2$ the following new constraints:

$$\sum_{j=1}^{i-1} v_{jt} \geq z_{it} \qquad \forall t \in T, i \in I \setminus \{1\} \tag{32}$$

Constraints (32) imposes that if there is a full 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 (32) will hence assign the setup crossover to the item with the highest active setup time.

4 Analysis of the Formulations

4.1 Theoretical Analysis of the Formulations

In this section we prove the relationship among the optimal objective function values for all of the discussed formulations and we show that we can always construct an alternative solution with the same objective function value imposing the setup crossover for the product with the highest active setup time. Note that $v(F)$ indicates the optimal objective function value of formulation F and $S(F)$ denotes the set of feasible solutions.

Lemma 1 $v(F0) \geq v(F3) \geq v(F4) \geq v(F5) = v(F1) = v(F2)$.

Proof: The first inequality ($v(F0) \geq v(F3)$) is trivial, since by adding the setup crossover the flexibility is increased and better solutions can be found. Moreover, it is clear that $S(0) \subseteq S(3)$ because by fixing $u_t = 0 \ \forall t \in T$ in formulation $F3$ we obtain the classical formulation $F0$.

Formulation $F3$ is more restrictive than formulation $F4$. Comparing the right-hand side of constraints (27) and (29) we have that $\min_{\forall j \in I}\{st_{jt}\} \leq st_{it}y_{it} + \max_{\forall j \in I}\{st_{jt}\}(1-y_{it}), \forall i \in I$. Therefore $S(3) \subseteq S(4)$, which proves the proposed second relationship ($v(F3) \geq v(F4)$).

We also see that ($S(4) \subseteq S(5)$). The reason is that in $F5$ the allowable time for a setup crossover is restricted to the maximum of the active setups, whereas in $F4$ it is restricted to the minimum of the active setups, which is more restrictive.

To show that $v(F5) = v(F1)$ 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.

Finally, the formulations $F1$ and $F2$ are both valid for the same problem and hence provide the same optimal objective function value ($v(F1) = v(F2)$).

Lemma 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. $st_{i',t+1} \geq st_{i,t+1}$ and $y_{i',t+1} = 1$). This solution can be constructed as follows:

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

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.

Lemma 3 Given a feasible (or optimal) solution for $F2$, with a setup crossover for product i from period t to $t+1$ (i.e. $v_{i,t+1} = 1$, $v_{j,t+1} = 0 \; \forall j \in I \setminus \{i\}$ 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. $st_{i',t+1} \geq st_{i,t+1}$ and $z_{i',t+1} = 1$). This solution can be constructed as follows:

$$v_{i',t+1} = 1, \; z_{i',t+1} = 0$$
$$v_{i,t+1} = 0, \; z_{i,t+1} = 1$$
$$l_{i't} \longleftarrow l_{it}, \; l_{it} = 0$$
$$f_{i',t+1} = st_{i',t+1} - l_{i't}, \; f_{i,t+1} = 0$$

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 $t+1$ has the same value after the changes (see Table 1):

	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	$st_{i',t+1} + st_{i,t+1} - l_{it}$	$st_{i,t+1} + st_{i',t+1} - l_{i't}$

Table 1: Constraints (17) for period $t+1$ and items i and i'.

(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.

4.2 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 non-uniform periods. Tables 2 and 3 contain the parameters, the demand and capacity values. Note that the parameters are time independent.

	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 2: Parameters for the example.

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

Table 3: Demand and capacity data.

All formulations were solved to optimality using this data set. Figure 1 and Table 5 illustrate the graphical solutions and the relevant non-zero variable values, respectively.

In Figure 1, white blocks represent production time that is consumed in that period, dark grey represents the setup time and light grey represents idle time. The values of the non-zero decision variables can be found in Table 5. 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.

Although slightly different, the solutions found by the formulations $F1$, $F1'$, $F5$, $F2$ and $F2'$ 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 $F5$, $F2$ and $F2'$ the setup for this item is completely done in period 2. Observe that the solution obtained by $F2$, $F2'$ and $F5$ can directly be transformed into the solution obtained by $F1$ and

$F1'$ by splitting the setup of product A over periods 1 and 2. The two solutions presented are both feasible for $F1$, $F1'$, $F2$, $F2'$ and $F5$. We see hence that there can be equivalent alternative optimal solutions. Note that there are no inventories in the solutions obtained by these formulations. Note also that although formulations $F2$ and $F2'$ presenting the same solution, the value of some variables are different (see Table 4). It occurs because for a product with a partial setup between periods $t-1$ and t ($v_{it} = 1$), it is still possible in some cases to do the complete setup in period t by choosing $l_{i,t-1} = 0$ or in period $t-1$ by choosing $f_{it} = 0$.

For the formulation $F3$ the maximum extra time that could be borrowed in each period was 1. Consequently, the optimal solution contains 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 (30 units). The optimal objective function value for this formulation is 574. When we limit the borrowed extra time to the minimum of the active setup times ($F4$), the formulation has more flexibility to find better solutions than $F3$. The optimal solution found by formulation $F4$ has inventory only for item C from period 2 to period 3 (30 units) and the objective function value of the optimal solution is 52. The results are in line with the relationships proposed in Lemma 1, and indicate that the inequalities can be strict.

5 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, 2.27GHz with 6GB of RAM and the Windows operating system. The computational tests involve four experiments based on standard instances proposed in Trigeiro et al. (1989). In the first experiment, the formulations are tested for the well know F and G instances. In the second experiment, the formulations were tested on a large data set of 540 standard instances, in the third one we test some adapted instances with high values for the inventory costs and finally, in the fourth one we test some instances considering the possibility of backlog. We have limited the computational time in all experiments to 1800 seconds per instance. Note that in these instances the unit production costs are not considered.

5.1 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. 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.

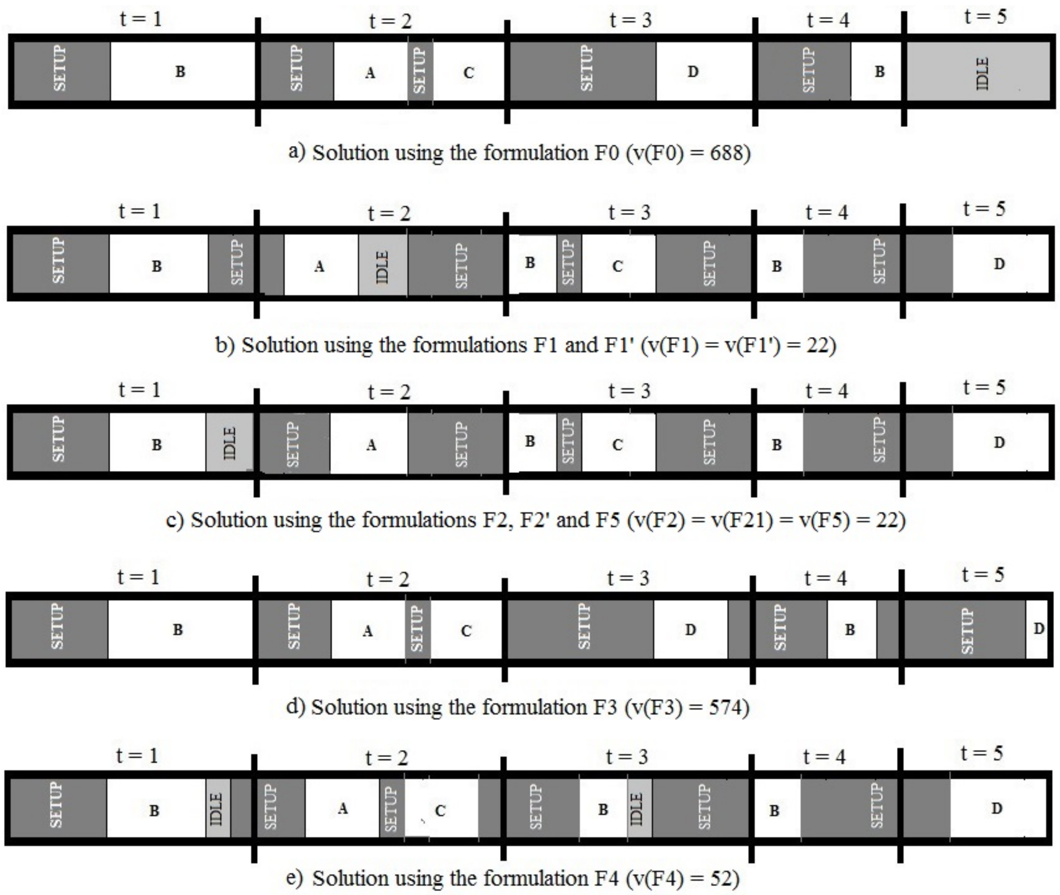

Figure 1: Graphical solution of the example for all formulations.

In Table 5 we give the upper bounds (Columns UP) 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 than 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 the formulation $F2$ and $F2'$ for some instances are faster than the classical formulation.

We observe that $F1$ is much faster than $F0$, which is surprising since it contains more binary variables. Omitting the binary condition on the v_{it} variables as done in formulation $F5$ does not result in a significant change in the CPU time. It only provides a very small decrease compared to $F1$. We observe that $F3$ and $F4$, i.e. the two restricted models without binary variables to indicate the crossover take significantly more time to be solved compared to formulations $F1$ and $F5$.

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

$F0$ 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						
$x_{A22}=1$	$x_{B11}=1$	$x_{B33}=1$	$x_{B44}=1$	$x_{C33}=1$	$x_{D55}=1$	
$z_{A2}=1$	$z_{B1}=1$	$z_{C3}=1$				
$v_{B3}=1$	$v_{B4}=1$	$v_{D5}=1$				
$l_{B2}=4$	$l_{B3}=4$	$l_{D4}=4$				
$f_{D5}=2$						
$F2'$ variables						
$x_{A22}=1$	$x_{B11}=1$	$x_{B33}=1$	$x_{B44}=1$	$x_{C33}=1$	$x_{D55}=1$	
$z_{C3}=1$						
$v_{A2}=1$	$v_{B1}=1$	$v_{B3}=1$	$v_{B4}=1$	$v_{D5}=1$		
$l_{B2}=4$	$l_{B3}=4$	$l_{D4}=4$				
$f_{A2}=3$	$f_{B1}=4$	$f_{D5}=2$				
$F3$ variables						
$x_{A22}=1$	$x_{B11}=1$	$x_{B13}=1$	$x_{B44}=1$	$x_{C23}=1$	$x_{D35}=0.75$	$x_{D55}=0.25$
$y_{A2}=1$	$y_{B1}=1$	$y_{B4}=1$	$y_{C2}=1$	$y_{D3}=1$	$y_{D4}=1$	
$u_3=1$	$u_4=1$					
$F4$ variables						
$x_{A22}=1$	$x_{B11}=1$	$x_{B33}=1$	$x_{B44}=1$	$x_{C23}=1$	$x_{D55}=1$	
$y_{A2}=1$	$y_{B1}=1$	$y_{B3}=1$	$y_{B4}=1$	$y_{C2}=1$	$y_{D5}=1$	
$u_1=1$	$u_2=1$	$u_3=4$	$u_4=4$			
$F5$ 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 4: Variables values of example for all formulations.

formulation $F2$ the symmetry breaking constraints are very efficient and the difference of CPU times between the formulations $F2$ and $F2'$ are very significant.

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 $F1$, $F1'$, $F2'$, $F4$ and $F5$ the solver has proven the optimality for all instances within the time limit. Using formulations $F0$, $F2$ and $F3$ the solver has proven the optimality for 98.6%, 88.9% and 99.3%, respectively.

5.2 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

	F		G6-15		G12-15		G24-15		G6-30		G12-30		G24-30	
Model	UP	Time	UP	Time	UP	Time	UP	Time	UP	Time	UP	Time	UP	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	99.72	1.6	99.65	8.6	99.95	3.9	99.93	8.3	99.73	81.1	99.92	411.3	99.99	446.6
F4	99.67	1.3	99.54	10.7	99.95	10.0	99.90	15.7	99.73	237.5	99.90	365.6	99.99	390.0
F5	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 5: Average general results for F and G data sets.

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).

Tables 6 to 10 show the overall performance of the formulations. We report the relative upper bounds (UP), computational times in seconds ($Time$) and percentage of instances solved to optimality within the limit of 1800 seconds (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 Tables 6 to 10 confirm the tendencies observed in Table 5. $F0$ is slower than $F1$, $F3$, $F4$ (except for 20 items) and $F5$. $F2$ is overall the slowest formulation and the performance of $F2'$ is in fact significantly better than $F2$. We see that $F1$ and $F5$ generally provide a similar performance (both in terms of CPU times and the percentage of optimal solutions found). $F3$ and $F4$ provide a significantly worse performance compared to $F1$ and $F5$. 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 decrease is 0.59% 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 $F5$ have the same optimal solutions, $F5$ found slightly better solutions for problems with 20 and 30 items. 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 96.1 and 9.5, respectively.

Table 7 shows that the capacity utilization is an important factor for the quality of solutions using setup crossover and the difficulty of the problems. For problems with loose capacity, the inclusion of a setup crossover is not so important. The problems are quite easy considering that the computational times are low and the percentage of solutions solved to optimality (OS) is very high. When the capacity is tight, the importance of including setup crossover and the difficulty of the problems increase. Note that the computational times are very high and the

	10 items			20 items			30 items			Aver. OS	
Model	UP	Time	OS	UP	Time	OS	UP	Time	OS	UP	Time
F0	100	589.8	70.6	100	684.9	63.9	100	707.1	62.8	100	10.7
F1	99.41	462.0	80.6	99.78	625.2	68.9	99.86	613.0	68.9	99.88	4.7
F2	99.43	813.7	57.2	99.79	859.5	53.9	99.89	865.3	55.0	99.88	96.1
F2'	99.43	628.0	68.8	99.79	685.0	64.5	99.89	672.8	65.5	99.88	9.5
F3	99.75	530.0	75.0	99.90	668.9	65.0	99.94	668.6	66.1	99.95	5.9
F4	99.73	539.1	73.4	99.90	688.3	63.9	99.94	666.8	65.6	99.94	6.3
F5	99.41	475.9	78.3	99.77	604.5	70.0	99.83	609.9	68.3	99.88	4.6

Table 6: General average results aggregated per number of items.

percentage of instances that CPLEX solved to optimality is very low for these instances.

	Loose Capacity			Normal Capacity			Tight Capacity		
Model	UP	Time	OS	UP	Time	OS	UP	Time	OS
F0	100	29.3	98.9	100	404.8	84.4	100	1547.8	15.0
F1	99.95	17.2	99.4	99.79	280.7	91.1	99.30	1402.3	27.2
F2	99.95	126.2	96.1	99.79	777.3	61.7	99.37	1634.9	9.45
F2'	99.95	27.1	99.4	99.79	435.7	81.6	99.37	1520.4	17.8
F3	99.98	25.7	98.9	99.91	347.8	87.2	99.70	1493.9	20.0
F4	99.98	25.7	98.9	99.90	354.9	85.6	99.69	1513.7	18.3
F5	99.95	16.0	99.4	99.79	276.9	90.6	99.28	1397.4	26.7

Table 7: General average results aggregated per capacity.

Table 8 presents the results aggregated per setup cost level. The benefits of including a setup crossover decreases when the value of setup cost increases. It occurs because if the setup cost is high, the formulations try to reduce the numbers of setups and keep more items in inventory whereas one of the main gains of setup crossover is exactly the flexibility to produce as close as possible to the demand period avoiding big quantities of inventory. Note also that computational times increase and the OS decrease significantly when the value of setup cost increases.

	Low setup cost			Normal setup cost			High setup cost		
Model	UP	Time	OS	UP	Time	OS	UP	Time	OS
F0	100	477.8	74.4	100	623.3	67.2	100	880.8	57.2
F1	99.55	384.1	81.7	99.75	521.2	73.3	99.74	764.9	62.8
F2	99.56	550.4	70.0	99.76	772.8	60.0	99.79	1215.2	36.1
F2'	99.56	470.6	75.0	99.76	572.4	69.4	99.79	858.5	58.8
F3	99.82	440.9	78.3	99.89	594.6	68.3	99.88	832.1	60.6
F4	99.82	455.6	76.7	99.88	600.6	68.3	99.87	828.1	59.4
F5	99.52	378.6	81.1	99.74	545.3	73.3	99.74	766.4	62.8

Table 8: General average results aggregated per setup cost level.

Tables 9 and 10 show the results taking into account the setup time and demand variability, respectively. In these tables we can see that with increasing setup times and demand variability, the computational times decrease, OS increases and the relative upper bounds decrease, indicating a larger benefit provided by the crossover. Note that for problems with low setup times and low demand variability the proposed formulation $F5$ found again slightly better solutions than the formulations from the literature $F1$ and $F2$.

	Low setup time			High setup time		
Model	UP	Time	OS	UP	Time	OS
F0	100	715.6	64.8	100	605.7	69.6
F1	99.78	669.9	68.1	99.58	463.5	77.8
F2	99.80	931.3	51.4	99.60	761.0	59.6
F2’	99.80	764.7	65.5	99.60	561.8	72.2
F3	99.91	699.1	63.7	99.81	545.9	73.7
F4	99.91	710.9	62.6	99.80	551.9	72.9
F5	99.76	670.1	66.3	99.58	456.8	78.1

Table 9: General average results aggregated per setup time.

	Medium demand variability			High demand variability		
Model	UP	Time	OS	UP	Time	OS
F0	100	728.2	62.6	100	593.1	70.0
F1	99.72	645.9	67.8	99.64	487.6	77.0
F2	99.75	844.4	55.2	99.65	847.9	55.6
F2’	99.75	735.7	63.3	99.65	585.5	71.8
F3	99.88	702.1	63.7	99.85	542.9	73.7
F4	99.88	704.2	63.3	99.83	558.6	72.2
F5	99.70	644.7	67.0	99.64	482.1	77.4

Table 10: General average results aggregated per demand variability.

Aiming to do a further analysis of the effect of introducing a setup crossover, Table 11 shows the behavior 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 through the increased flexibility. This might require, however, a slight increase in the number of setups as well.

		Model F0				Model F1			
		SC(%)	HC(%)	setup	inv.	SC(%)	HC(%)	setup	inv.
Capacity	Loose	74.75	25.24	101.7	10931	74.78	25.21	101.6	10880
	Normal	72.87	27.13	101.9	11133	73.04	26.95	101.9	11120
	Tight	62.81	37.19	95.3	12452	63.87	36.17	95.7	12158
S. cost	Low	81.63	18.37	151	2051	82.56	17.44	151.6	1932
	Normal	65.33	34.67	90.4	9232	65.64	34.36	90.4	9118
	High	63.46	36.54	57.6	23234	63.47	36.53	57.2	23108
S. time	Low	69.05	30.95	98.8	11516	69.36	30.64	98.8	11467
	High	71.23	28.77	100.6	11435	71.75	28.25	100.7	11305
Demand	Medium	72.07	27.93	108.3	11516	72.30	27.70	108.2	11485
	High	68.21	31.79	91.1	11495	68.81	31.19	91.3	11286
Average		**70.14**	**29.86**	**99.6**	**11505**	**70.56**	**29.44**	**99.7**	**11386**

Table 11: Detailed results for 10 items.

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. Tables A.20 and A.21 in the appendix contain the results for all 108 different combinations considering the five characteristics discussed in this experiment. Note that for each of the 108 combinations, 5 instances were tested. We observe 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 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 aggregated analysis presented in Tables 6 to 10.

5.3 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 12 and 13 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 $F5$ have a similar performance. $F3$ and $F4$ are slower compared to $F1$ and $F5$, but faster than $F0$. $F2$ 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 almost 3% (for the inventory cost $\times$ 10) and 8% (for the inventory cost $\times$ 100).

Table 12 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 (4.2% cost decrease), high setup times (4.4% cost decrease) and low setup cost (4.5% cost decrease). The analysis further reveals that the approximate formulations $F3$ and $F4$ are not able to capture all of the benefits with an average cost decrease of 1.3%.

		Model F0		Model F1		Model F2		Model F2'		Model F3		Model F4		Model F5	
		UP	T(s)	UP	T(s)	UP	T(s)	UP	T(s)	UP	T(s)	UP	T(s)	UP	T(s)
Capacity	Loose	100	81	99.00	2	99.00	185	99.00	66	99.54	46	99.53	43	99.00	2
	Normal	100	499	96.40	342	96.41	926	96.41	579	98.42	363	98.37	393	96.40	331
	Tight	100	1313	95.79	1150	95.82	1683	95.79	1428	98.25	1240	98.12	1292	95.78	1138
S. cost	Low	100	292	95.49	225	95.49	604	95.49	436	98.19	265	98.16	268	95.49	225
	Normal	100	727	97.05	485	97.06	1036	97.04	758	98.58	574	98.54	619	97.04	473
	High	100	875	98.64	785	98.67	1154	98.67	880	99.44	811	99.33	842	98.64	775
S. time	Low	100	496	98.53	468	98.55	842	98.54	553	99.46	490	99.42	524	98.53	461
	High	100	766	95.60	528	95.61	1021	95.59	719	98.01	610	97.93	628	95.58	521
Demand	Medium	100	811	97.54	667	97.55	944	97.54	872	98.88	719	98.83	725	97.53	666
	High	100	451	96.59	330	96.60	920	96.59	511	98.59	381	98.52	427	96.59	316
Average		**100**	**631**	**97.06**	**498**	**97.08**	**932**	**97.07**	**691**	**98.74**	**550**	**98.67**	**576**	**97.06**	**491**

Table 12: Average general results with inventory costs multiplied by 10.

Table 13 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 6.3%.

		Model F0		Model F1		Model F2		Model F2'		Model F3		Model F4		Model F5	
		UP	T(s)	UP	T(s)	UP	T(s)	UP	T(s)	UP	T(s)	UP	T(s)	UP	T(s)
Capacity	Loose	100	125	93.71	34	93.72	216	93.71	129	97.22	72	97.21	80	93.71	34
	Normal	100	514	89.62	332	89.64	931	89.62	556	95.20	395	95.13	408	89.62	309
	Tight	100	1168	94.03	1016	94.06	1683	99.04	1365	97.49	1068	97.39	1274	94.02	992
S. cost	Low	100	290	89.82	222	89.83	604	89.82	384	95.65	246	95.61	268	89.82	214
	Normal	100	593	92.04	399	92.06	1037	92.04	740	96.12	454	96.08	633	92.04	382
	High	100	924	95.50	760	95.54	1189	95.50	927	98.14	834	98.04	860	95.49	740
S. time	Low	100	399	95.88	389	95.90	835	95.89	595	98.44	387	98.42	500	95.88	382
	High	100	805	89.02	532	89.04	1052	89.02	772	94.83	636	94.73	675	89.02	508
Demand	Medium	100	787	93.31	658	93.35	970	93.32	892	96.67	691	96.63	754	93.31	635
	High	100	417	91.59	263	91.60	917	91.59	475	96.60	332	96.52	421	91.59	255
Average		**100**	**602**	**92.45**	**460**	**92.47**	**943**	**92.45**	**683**	**96.63**	**512**	**96.57**	**587**	**92.45**	**445**

Table 13: Average general results with inventory costs multiplied by 100.

Table 14 and 15 show the behavior of the solutions for the results with inventory costs multiplied by 10 and 100. We observe that, contrary to the results of Table 11, 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 (62.74% for formulation $F0$ and 59.89% 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 14) the total inventory goes down by approximately 5%, and the total setups only increase by 0.5%. Regarding the instances in which the inventory costs are multiplied by 100 (Table 15) the total inventory goes down by approximately 6% and the total setups only increase by 0.5%.

		Model F0				Model F1			
		SC(%)	HC(%)	setup	inv.	SC(%)	HC(%)	setup	inv.
Capacity	Loose	88.33	11.66	158.8	1312	89.64	10.35	159.4	1200
	Normal	64.39	35.60	145.4	3085	66.88	33.11	146.3	2868
	Tight	25.25	74.74	113.3	9260	26.51	73.48	111.8	8881
S. cost	Low	62.99	37.00	163.7	1603	65.26	34.73	164.4	1461
	Normal	57.04	42.95	139.0	4152	58.89	41.10	139.8	3901
	High	57.95	42.04	112.8	7902	58.88	41.11	113.1	7588
S. time	Low	61.42	38.57	141.2	4144	62.16	37.83	141.5	4031
	High	57.23	42.76	135.8	4961	59.86	40.13	136.8	4602
Demand	Medium	64.62	35.37	147.3	4198	66.01	33.98	148.0	3972
	High	54.03	45.96	129.7	4907	56.01	43.98	130.3	4661
Average		**59.33**	**40.66**	**138.5**	**4552**	**61.01**	**38.98**	**139.1**	**4316**

Table 14: General detailed results with inventory costs multiplied by 10.

		Model F0				Model F1			
		SC(%)	HC(%)	setup	inv.	SC(%)	HC(%)	setup	inv.
Capacity	Loose	74.52	25.48	170.4	712	78.69	21.31	170.9	597
	Normal	31.18	68.82	152	2810	35.21	64.79	153	2562
	Tight	6.06	93.94	114.9	9316	6.43	93.57	115.4	8905
S. cost	Low	45.49	54.51	164.9	1602	49.15	50.85	165.8	1458
	Normal	32.59	67.41	143.5	4144	35.86	64.14	144.0	3877
	High	33.69	66.31	128.8	7093	35.32	64.68	129.5	6728
S. time	Low	43.84	56.16	150.9	3768	45.76	54.24	151.2	3641
	High	30.67	69.33	140.7	4791	34.46	65.54	141.7	4022
Demand	Medium	47.47	52.53	152.6	3893	50.54	49.46	153.0	3659
	High	27.04	72.96	138.9	4666	29.68	70.32	139.8	4384
Average		**37.26**	**62.74**	**145.8**	**4280**	**40.11**	**59.89**	**146.5**	**4021**

Table 15: General detailed results with inventory costs multiplied by 100.

5.4 Results for Experiment 4

In this experiment the formulations were adapted to allow backlog and were tested on a set of 60 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 16 to 19 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 16 to 19 show 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 16 and 17 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 increase. 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 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.

		Model F0			Model F1		
		UP	T(s)	FS	UP	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	Low	100	1800	93	97.86	1617	97
	High	100	1467	93	97.62	1350	93
Average		**100**	**1633**	**93**	**97.73**	**1483**	**95**

Table 16: General results with backlog and 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	15345	10	48.84	49.17	1.97	78.2	14766	10
	H	43.25	21.02	35.72	60.3	21167	793	44.71	21.48	33.80	60.3	20921	759
S. time	L	40.98	51.43	7.57	83.5	14911	252	41.55	50.92	7.51	83.7	14639	252
	H	49.63	36.51	13.85	88.0	13714	211	51.71	35.46	12.82	88.8	13246	192
Demand	M	44.44	41.86	13.69	90.3	14143	341	45.73	41.08	13.18	90.9	13739	335
	H	46.79	45.18	8.18	81.6	14397	120	48.25	44.19	7.54	81.9	14047	106
Average		**45.61**	**43.44**	**10.93**	**85.9**	**14270**	**230**	**46.99**	**42.64**	**10.36**	**86.4**	**13893**	**221**

Table 17: General detailed results with backlog capacity reduced by 5%.

Tables 18 and 19 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 more relevant (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.

		Model F0			Model F1		
		UP	T(s)	FS	UP	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 18: General results with backlog capacity reduced by 10%.

6 Conclusion

In this paper, the lot sizing problem with capacity constraints and setup crossover was studied. A reformulation of the problem using the simple plant location model was used. Three new

		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	12399	781	29.01	47.58	23.39	92.4	12083	775
	N	35.58	50.34	14.06	65.3	19995	1152	36.82	49.38	13.79	65.3	19021	959
	H	3.08	3.86	93.04	33.2	28080	7925	3.38	4.03	92.57	33.4	26682	6977
S. time	L	9.10	39.31	51.58	66.9	19588	2525	9.25	39.43	51.30	66.8	19164	2324
	H	34.41	44.67	20.91	78.9	15714	1606	36.02	43.25	20.72	80.0	14940	1415
Demand	M	22.78	39.22	37.99	73.5	18012	2625	23.68	38.61	37.69	73.9	17205	2299
	H	31.55	47.80	20.64	77.4	15505	964	33.03	46.43	20.52	78.6	15012	935
Average		**26.68**	**43.03**	**30.28**	**75.2**	**16898**	**1886**	**27.84**	**42.08**	**30.06**	**76.0**	**16231**	**1693**

Table 19: General detailed results with backlog capacity reduced by 10%.

formulations avoiding the necessity to define new extra binary variables to model the setup crossover and two adding new symmetry breaking constraints were proposed. Using CPLEX 12.6 these formulations were compared with two different formulations proposed in literature to model the setup crossover using extra binary variables and with the classical assumption where a setup crossover is not allowed. The results show that the proposed formulations are efficient, specially the formulation $F5$ which is slightly better than the formulation $F1$ proposed in the literature. Comparing the benefits obtained allowing a setup crossover with the classical assumption, we conclude that it depends on the characteristic of the problem, but especially for problems with high inventory cost it can be very significant. Indeed, when the inventory costs were multiplied by 10 and 100 the cost decrease obtained by introducing the possibility of a setup crossover are on average 3% and 7.5%. Finally, we also conclude that the benefits obtained allowing a setup crossover are 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 2011/22647-0) and by the Natural Sciences and Engineering Research Council of Canada [371966-2009]. This support is gratefully acknowledged.

A Table with the 108 combinations of experiment 2

Tables A.20 and A.21 show all 108 possible combinations including the five characteristics analyzed in experiment 2. Observe that the results are organized according to increasing upper bound.

Characteristic of the problem					Formulation F5		
Items	Cap.	S. Cost	S. Time	Demand	UB	Time	OS
10	T	L	H	M	97.23	864.8	60
10	T	L	H	H	97.38	2.7	100
30	T	L	L	M	98.26	1800	0
10	T	H	H	H	98.34	1800	0
10	T	L	L	M	98.53	1561.7	20
10	T	M	H	H	98.59	661.4	80
10	T	H	L	M	98.65	1800	0
10	T	L	L	H	98.71	618.9	80
10	M	L	H	H	98.71	0.3	100
10	T	M	H	M	98.86	1714.5	20
10	T	H	H	M	98.88	1440.0	20
20	T	L	H	H	99.02	72.8	100
20	T	L	H	M	99.12	1226.4	60
20	T	M	H	H	99.29	1211.2	60
20	T	H	H	M	99.30	1800	0
10	T	M	L	H	98.32	1800	0
30	T	H	H	M	99.39	1800	0
10	M	H	H	H	99.40	223.6	100
20	T	L	L	H	99.40	1445.5	20
20	T	H	H	H	99.41	1800	0
10	T	M	L	H	99.42	112.0	60
30	T	L	H	M	99.44	1494.5	20
20	T	M	H	M	99.45	1800	0
10	T	H	L	H	99.46	1690.9	20
20	T	L	L	M	99.47	1800	0
10	M	L	L	M	99.48	0.6	100
30	T	L	L	H	99.53	1800	0
10	M	H	H	M	99.59	332.6	100
30	T	L	H	H	99.60	14.5	100
10	M	M	H	H	99.61	1.4	100
30	T	M	H	M	99.62	1800	0
30	T	H	H	H	99.63	1800	0
10	M	L	H	M	99.66	0.2	100
30	T	M	H	H	99.70	842.4	60
20	T	M	L	H	99.72	1800	0
20	T	H	L	M	99.73	1800	0
20	T	M	L	H	99.73	1800	0
10	M	M	H	M	99.74	2.26	100
10	M	H	L	H	99.74	48.7	100
20	M	L	H	H	99.74	0.7	100
10	L	H	H	H	99.76	2.6	100
30	T	H	L	H	99.77	1800	0
10	T	H	L	M	99.78	1800	0
10	M	M	L	H	99.79	16.3	100
10	L	H	H	M	99.82	6.1	100
30	T	M	L	H	99.82	1800	0
20	M	H	H	H	99.83	1.6	100
20	M	L	L	H	99.84	2.4	100
20	M	H	H	M	99.84	847.6	60
20	T	H	L	H	99.84	1800	0
30	T	M	L	M	98.84	1800	0
10	L	M	L	H	99.85	0.6	100

Table 20: General average results aggregated.

References

M. A. F. Belo-Filho. B. Almada-Lobo. F. M. B. Toledo, Models for capacitated lot-sizing problem with backlogging. setup carryover and crossover, *Journal of the Operational Research Society*, in press.

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

Characteristic of the problem					Formulation F5		
Items	Cap.	S. Cost	S. Time	Demand	UB	Time	OS
10	L	M	H	H	99.85	0.6	100
10	L	L	L	H	99.86	0.2	100
20	M	H	H	H	99.86	85.6	100
10	L	L	H	H	99.88	0.1	100
10	M	M	L	M	99.89	5.7	100
10	L	M	H	M	99.89	0.6	100
30	M	L	H	H	99.89	0.8	100
30	M	L	L	H	99.91	1.6	100
30	M	H	H	H	99.91	368.6	100
10	M	H	L	M	99.92	836.3	80
10	L	H	L	H	99.92	8.7	100
20	M	M	H	M	99.92	4.0	100
30	M	M	H	H	99.92	14.5	100
20	M	L	H	M	99.93	0.3	100
20	M	M	L	H	99.93	21.2	100
20	L	H	H	H	99.94	6.4	100
20	L	M	H	H	99.94	0.6	100
10	M	L	L	M	99.95	0.2	100
20	M	H	L	M	99.95	1247.3	40
20	M	H	L	H	99.95	584.5	80
30	M	H	H	M	99.95	854.7	60
20	L	L	H	H	99.96	0.3	100
20	L	H	H	M	99.96	17.4	100
30	M	M	H	M	99.96	8.2	1000
30	M	H	L	H	99.96	270.3	100
30	L	H	H	H	99.96	5.2	100
10	L	M	L	M	99.97	0.7	100
10	L	H	L	M	99.97	15.5	100
20	M	M	L	M	99.97	11.5	100
20	L	L	L	H	99.97	0.5	100
20	L	H	L	H	99.97	5.4	100
30	M	H	L	M	99.97	1369.1	40
30	L	H	H	M	99.97	9.2	100
20	L	M	L	H	99.98	0.5	100
20	L	M	L	M	99.98	1.1	100
20	L	M	H	H	99.98	0.7	100
20	L	H	L	M	99.98	16.8	100
30	M	L	H	M	99.98	0.4	100
30	M	M	L	M	99.98	33.4	100
30	M	M	L	H	99.98	20.1	100
30	L	M	H	H	99.99	7.3	100
20	M	L	L	M	99.99	0.4	100
30	L	M	L	M	99.99	0.6	100
30	L	M	H	M	99.99	1.0	100
30	L	H	L	M	99.99	15.7	100
30	L	L	L	H	99.99	0.3	100
30	L	M	L	H	99.99	1.1	100
30	L	H	L	H	99.99	10.4	100
10	L	L	L	M	100	0.1	100
10	L	L	H	M	100	0.1	100
20	L	L	L	M	100	0.2	100
20	L	L	H	M	100	0.2	100
30	L	L	L	M	100	0.3	100
30	L	L	H	M	100	0.3	100
30	L	L	H	H	100	0.3	100
30	M	L	L	M	100	0.3	100

Table 21: General average results aggregated.

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

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

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

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

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

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

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

G. M. Kopanos. L. Puigjaner. C. T. Maravelias, Production Planning and Scheduling of Parallel Continuous Processes with Product Families. *Industrial and Engineering Chemistry Research*, **50(3)** (2011), 1369–1378.

J. Krarup. O. Bilde, 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*, (1977), 155–186.

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

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

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

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

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

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

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

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

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

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