



**HAL**  
open science

## Multi-product valid inequalities for the discrete lot-sizing and scheduling problem

Céline Gicquel, Michel Minoux

► **To cite this version:**

Céline Gicquel, Michel Minoux. Multi-product valid inequalities for the discrete lot-sizing and scheduling problem. *Computers and Operations Research*, 2015, 54, pp.12-20. 10.1016/j.cor.2014.08.022 . hal-01262267

**HAL Id: hal-01262267**

**<https://hal.science/hal-01262267>**

Submitted on 27 Jan 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Multi-product valid inequalities for the discrete lot-sizing and scheduling problem

C. Gicquel<sup>a,\*</sup>, M. Minoux<sup>b</sup>

<sup>a</sup> *Laboratoire de Recherche en Informatique  
Université Paris Sud  
Campus d'Orsay, bâtiment 650  
91405 Orsay Cedex, France*

<sup>b</sup> *Laboratoire d'Informatique de Paris 6  
Université Pierre et Marie Curie  
4 place Jussieu  
75005 Paris, France*

---

## Abstract

We consider a problem arising in the context of industrial production planning, namely the multi-product discrete lot-sizing and scheduling problem with sequence-dependent changeover costs. We aim at developing an exact solution approach based on a Cut & Branch procedure for this combinatorial optimization problem. To achieve this, we propose a new family of multi-product valid inequalities which corresponds to taking into account the conflicts between different products simultaneously requiring production on the resource. We then present both an exact and a heuristic separation algorithm which form the basis of a cutting-plane generation algorithm. We finally discuss computational results which confirm the practical usefulness of the proposed inequalities at strengthening the MILP formulation and at reducing the overall computation time.

*Keywords:* production planning, discrete lot-sizing problem, mixed-integer linear programming, valid inequalities

---

\*Corresponding author: Céline Gicquel; Tel: +33 (0)1 69 15 42 26; E-mail: celine.gicquel@lri.fr

## 1. Introduction

We consider an optimization problem arising in the context of industrial production planning, namely a lot-sizing problem. Lot-sizing arises in production planning whenever changeover operations such as preheating, tool changing or cleaning are required between production runs of different products on a machine. The amount of the related changeover costs usually does not depend on the number of products processed after the changeover. Thus, to minimize changeover costs, production should be run using large lot sizes. However, this generates inventory holding costs as the production cannot be synchronized with the actual demand pattern: products must be held in inventory between the time they are produced and the time they are used to satisfy customer demand. The objective of lot-sizing is thus to reach the best possible trade-off between changeover and inventory holding costs while taking into account both the customer demand satisfaction and the technical limitations of the production system.

An early attempt at modelling this trade-off can be found in [19] for the problem of planning production for a single product on a single resource with an unlimited production capacity. Since this seminal work, a large part of the research on lot-sizing problems has focused on modelling operational aspects in more detail to answer the growing industry need to solve more realistic and complex production planning problems. An overview of recent developments in the field of modelling industrial extensions of lot-sizing problems is provided in [11].

In the present paper, we focus on one of the variants of lot-sizing problems mentioned in [11], namely the multi-product single-resource discrete lot-sizing and scheduling problem or DLSP. As defined in [7, 11], several key assumptions are used in the DLSP to model the production planning problem:

- A set of products is to be produced on a single capacitated production resource.
- A finite time horizon subdivided into discrete periods is used to plan production.
- Demand for products is time-varying (i.e. dynamic) and deterministically known.

- At most one product can be produced per period (small bucket model) and the facility processes either one product at full capacity or is completely idle (discrete or all-or-nothing production policy).
- Costs to be minimized are the inventory holding costs and the changeover costs.

In the DLSP, it is assumed that a changeover between two production runs for different products results in a changeover cost. Changeover costs can depend either on the next product only (sequence-independent case) or on the sequence of products (sequence-dependent case). We consider in the present paper the DLSP with sequence-dependent changeover costs (denoted DLSPSD in the sequel). Sequence-dependent changeover costs are mentioned in [11] as one of the relevant operational aspects to be incorporated into lot-sizing models. Moreover, a significant number of real-life lot-sizing problems involving sequence-dependent changeover costs have been recently reported in the academic literature: see for instance [4] for an injection moulding process, [17] for a textile fibre industry or [6] for soft drink production.

A wide variety of solution techniques from the Operations Research field have been proposed to solve lot-sizing problems: the reader is referred to [3, 10] for recent reviews on the corresponding literature. The present paper belongs to the line of research dealing with exact solution approaches, i.e. aiming at providing guaranteed optimal solutions to the problem. A large amount of existing solution techniques in this area consists in formulating the problem as a mixed-integer linear program (MILP) and in relying on a Branch & Bound type procedure to solve the obtained MILP. However the computational efficiency of such a procedure strongly depends on the quality of the lower bounds used to evaluate the nodes of the search tree. In the present paper, we seek to improve the quality of these lower bounds so as to decrease the total computation time needed to obtain guaranteed optimal solutions.

Within the last thirty years, much research has been devoted to the polyhedral study of lot-sizing problems in order to obtain tight linear relaxations and improve the corresponding lower bounds: see e.g. [15] for a general overview of the related literature. In particular, valid inequalities which reduce the volume of the linear relaxation solution space by cutting off irrelevant parts have been proposed for several variants. Inequalities to strengthen the Capacitated Lot Sizing Problem (CLSP) are thus proposed in [1, 13, 14]. Contributions focusing specifically on the Discrete Lot Sizing

Problem (DLSP) can be found in [2, 5, 8, 18]. However, the known inequalities mainly exploit the underlying single-product subproblems and thus fail at *capturing the conflicts between multiple products sharing the same resource capacity*. This leads in some cases to significant residual integrality gaps for multi-product instances. In the present paper, we propose a new family of multi-product multi-period inequalities which enables us to partially remedy this difficulty for the DLSPSD. We then discuss both an exact and a heuristic algorithm to solve the corresponding separation problem. To the best of our knowledge, this is one of the first attempts at proposing multi-product valid inequalities for discrete lot-sizing problems.

The main contributions of the present paper are thus twofold. First we introduce a new family of valid inequalities representing conflicts on multi-period time intervals between several products simultaneously requiring production on the available resource. Second we formulate the corresponding separation problem as a quadratic binary program and propose to solve it either exactly by relying on a quadratic programming solver or approximately through a variable depth search heuristic algorithm of Kernighan-Lin type (see [12]). The results of our computational results show that the proposed inequalities are efficient at strengthening the linear relaxation of the problem and at decreasing the overall computation time needed to obtain guaranteed optimal solutions of the DLSPSD.

The remainder of the paper is organized as follows. In Section 2, we recall the initial MILP formulation of the multi-product DSLPSD as well as the previously published inequalities for the underlying single-product subproblems. We then present in Section 3 the proposed multi-product inequalities and discuss in Section 4 both an exact and a heuristic algorithm to solve the corresponding separation problem. Computational results are provided in Section 5.

## 2. MILP formulation of the DLSPSD

In this section, we first recall the initial MILP formulation of the DLSPSD. We use the network flow representation of changeovers between products, which was discussed among others in [2], as this leads to a tighter linear relaxation of the problem. We then present the inequalities proposed in [18] to strengthen the underlying single-product subproblems.

### 2.1. Initial MILP formulation

We wish to plan production for a set of products denoted  $p = 1 \dots P$  to be processed on a single production machine over a planning horizon involving  $T$  periods indexed  $t = 1 \dots T$ . Product  $p = 0$  represents the idle state of the machine and period  $t = 0$  is used to describe the initial state of the production system.

Production capacity is assumed to be constant throughout the planning horizon. We can thus w.l.o.g. normalize the production capacity to one unit per period and apply a pretreatment on the original demand matrix resulting in a demand matrix containing only binary numbers (see [2, 7, 9]). We denote  $d_{pt}$  the demand for product  $p$  in period  $t$ :  $d_{pt} = 1$  in case there is a demand for product  $p$  in period  $t$  corresponding to producing  $p$  at full capacity in a period,  $d_{pt} = 0$  otherwise. Furthermore, we denote  $h_p$  the inventory holding cost per unit per period for product  $p$  and  $S_{pq}$  the sequence-dependent changeover cost to be incurred whenever the resource setup state is changed from product  $p$  to product  $q$ .

Using this notation, the DLSPSD can be seen as the problem of assigning at most one product to each period of the planning horizon while ensuring demand satisfaction and minimizing both inventory and changeover costs. We thus introduce the following binary decision variables:

- $y_{pt}$  where  $y_{pt} = 1$  if product  $p$  is assigned to period  $t$ , 0 otherwise.
- $w_{pqt}$  where  $w_{pqt} = 1$  if there is a changeover from product  $p$  to product  $q$  at the beginning of  $t$ , 0 otherwise.

This leads to the following MILP formulation denoted DLSPSD0 for the problem.

$$Z^* = \min \sum_{p=1}^P \sum_{t=1}^T h_p \sum_{\tau=1}^t (y_{p\tau} - d_{p\tau}) + \sum_{p,q=0}^P S_{p,q} \sum_{t=1}^{T-1} w_{p,q,t} \quad (1)$$

$$\sum_{\tau=1}^t y_{p\tau} \geq \sum_{\tau=1}^t d_{p\tau} \quad \forall p, \forall t \quad (2)$$

$$\sum_{p=0}^P y_{pt} = 1, \quad \forall t \quad (3)$$

$$y_{p,t} = \sum_{q=0}^P w_{q,p,t} \quad \forall p, \forall t \quad (4)$$

$$y_{p,t} = \sum_{q=0}^P w_{p,q,t+1} \quad \forall p, \forall t \quad (5)$$

$$y_{pt} \in \{0, 1\} \quad \forall p, \forall t \quad (6)$$

$$w_{p,q,t} \in \{0, 1\} \quad \forall p, \forall q, \forall t \quad (7)$$

The objective function (1) corresponds to the minimization of the inventory holding and changeover costs over the planning horizon.  $\sum_{\tau=1}^t (y_{p\tau} - d_{p\tau})$  is the inventory level of product  $p$  at the end of period  $t$ . Constraints (2) impose that the cumulated demand over interval  $[1, t]$  is satisfied by the cumulated production over the same time interval. Constraints (3) ensure that, in each period, the resource is either producing a single product or idle. Constraints (4)-(5) link setup variables  $y_{pt}$  with changeover variables  $w_{pqt}$  through equalities which can be seen as flow conservation constraints in a network (see e.g. [2]). They ensure that in case product  $p$  is setup in period  $t$ , there is a changeover from another product  $q$  (possible  $q = p$ ) to product  $p$  at the beginning of period  $t$  and a changeover from product  $p$  to another product  $q$  (possible  $q = p$ ) at the end of period  $t$ .

## 2.2. Single-product valid inequalities

We now recall the expression of the inequalities proposed by [18] for the single product DLSP. We denote  $D_{p,t,\tau}$  the cumulated demand for product  $p$  in the interval  $\{t, \dots, \tau\}$  and  $\Delta_{p,v}$  the  $v^{\text{th}}$  positive demand period for product  $p$ .  $\Delta_{p,D_{p,1,t}+v}$  is thus the period in which the  $v^{\text{th}}$  positive unit demand for product  $p$  after period  $t$  occurs.

$$\sum_{\tau=1}^t (y_{p\tau} - d_{p\tau}) \geq u - \sum_{v=1}^u \left[ y_{p,t+v} + \sum_{\tau=t+v+1}^{\Delta_{p,d_{p,1,t}+v}} \sum_{q \neq p} w_{q,p,\tau} \right] \quad \forall p, \forall t, \forall u \in [1, D_{p,t+1,T}] \quad (8)$$

The idea underlying constraints (8) is to compute a lower bound on the inventory level of a product  $p$  at the end of a period  $t$  ( $\sum_{\tau=1}^t (y_{p\tau} - d_{p\tau})$ ) by considering both the demands and the resource setup states for this product in the forthcoming periods  $\tau = t+1 \dots \Delta_{p,D_{p,1,t}+u}$ . The reader is referred to [18] for a full proof of validity for these inequalities. In the computation experiments to be presented in Section 5, we use a standard cutting-plane generation algorithm to strengthen the formulation DLSPSD0 by adding violated

inequalities of family (8). Since in practice the number of such inequalities is limited, the separation in this case is efficiently performed by enumeration. The resulting improved formulation is denoted DLSPSD1.

Constraints (8) can be understood as a way to strengthen the demand satisfaction constraints (2) by expressing in a more detailed way the need for each individual product to access the resource in order to satisfy its own demand on a given subinterval of the planning horizon. However, in the resulting DLSPSD1 formulation, the conflicts between different products simultaneously requiring production on the resource will only be handled by the single-period capacity constraints (3). In what follows, we propose to improve this representation of the conflicts between products by considering multi-period multi-product inequalities.

### 3. Multi-product valid inequalities for the DLSPSD

We now present the multi-period multi-product valid inequalities proposed to strengthen the linear relaxation of the DLSPSD.

#### 3.1. General expression of the multi-product inequalities

As stated above, the proposed inequalities aim at improving the representation of the interactions between the products competing for the scarce capacity of the production resource. In formulation DLSPSD1, these interactions are only managed on a period by period basis via the capacity constraints (3). On the contrary, the proposed inequalities consider this competition on a multi-period time interval denoted  $[1, \theta]$  in what follows. More precisely, the competition is expressed as an opposition between two disjoint subsets of products denoted  $SP$  and  $SD$ . The product set  $SP$  corresponds to products for which production in period  $t \in [1, \theta]$  is considered and that may consequently take up the resource capacity in this period. If this is to happen, the products of the other set,  $SD$ , will not have access to the resource in period  $t$  and will lose the corresponding production capacity. The purpose of inequalities (9) below is to manage this opposition by stating that, if one of the products in  $SP$  is assigned for production in  $t$ , then we should make sure that the cumulated demand over interval  $[1, \theta]$  for the products in the other set  $SD$  will fit in the remaining production capacity. The right hand side of inequalities (9) computes a tight upper bound  $\sum_{\tau=1}^{\theta} \tilde{C}_{\tau}$  of this production capacity:  $\tilde{C}_{\tau}$  represents an upper bound of  $C_{\tau}$ , the capacity



available in period  $\tau \in [1, \theta]$  for the products in  $SD$  in case period  $t$  is devoted to the production of one of the products in  $SP$ .

**Proposition 1.**

Let  $SP \subset \{0 \dots P\}$  and  $SD \subset \{0 \dots P\}$  be two disjoint subsets of products. Let  $t \in [1, T]$  and consider  $[1, \theta] \subset [1, T]$  a time interval including period  $t$ . For each period  $\tau \in [1, \theta]$ , we denote  $SD_\tau = \{q \in SD \mid \Delta_{q, D_{q,1,\theta}} \geq \tau\}$ . The following inequality is valid for the multi-product DLSPSD.

$$\left[ \sum_{q \in SD} D_{q,1,\theta} \right] \left[ \sum_{p \in SP} y_{pt} \right] \leq \sum_{\tau=1}^{\theta} \tilde{C}_\tau \quad (9)$$

where  $\tilde{C}_\tau$  is defined by:

$$\tilde{C}_\tau = \min \left( \sum_{q \in SD_\tau} y_{q,\tau}, \sum_{p \in SP} y_{p,t} \right) \text{ if } \tau \in [1; t-2] \cup [t+2, \theta] \quad (9a)$$

$$\tilde{C}_{t-1} = \sum_{q \in SD_{t-1}, p \in SP} w_{qpt} \quad (9b)$$

$$\tilde{C}_t = 0 \quad (9c)$$

$$\tilde{C}_{t+1} = \sum_{p \in SP, q \in SD_{t+1}} w_{pq,t+1} \quad (9d)$$

*Proof.* Let  $(y, w)$  be a feasible solution of the DLSPSD. We arbitrarily choose a period  $t$ , an interval  $[1, \theta]$  including  $t$  and two disjoint subsets of products  $SP$  and  $SD$  and show that the proposed inequality (9) is valid for the considered feasible solution.

We distinguish two main cases:

- Case 1:  $\sum_{p \in SP} y_{pt} = 0$

In this case, the left hand side of the inequality is equal to 0 whereas the right hand side is nonnegative. Inequality (9) is thus trivially valid.

- Case 2:  $\sum_{p \in SP} y_{pt} = 1$

In this case, the left hand side of inequality (9) is equal to the total cumulated demand over interval  $[1, \theta]$  for the products belonging to  $SD$ , i.e. to  $\sum_{q \in SD} D_{q,1,\theta}$ .

$\sum_{p \in SP} y_{pt} = 1$  means that period  $t$  is devoted to the production of one of the products in  $SP$  and thus cannot be used to satisfy the cumulated demand for products in  $SD$  as these two product subsets are disjoint. Hence  $(y, w)$  can be a feasible solution of the DLSPSD if and only if the total cumulated

production for products in  $SD$  over the remaining periods  $1 \dots t-1, t+1 \dots \theta$  is sufficient to satisfy the cumulated demand  $\sum_{q \in SD} D_{q,1,\theta}$ .

We now seek to compute a tight upper bound  $\tilde{C}_\tau$  for the production capacity  $C_\tau$  available in each period  $\tau \in [1, t-1] \cup [t+1, \theta]$  for the products in  $SD$ :

- By capacity constraints (3), we have  $C_\tau \leq 1$ , i.e.  $C_\tau \leq \sum_{p \in SP} y_{pt}$ .
- Moreover, the cumulated demand  $D_{q,1,\theta}$  for a product  $q \in SD$  can only be satisfied by a production for this product in period  $\tau$  if  $\tau$  is within the interval  $[1, \Delta_{q,D_{q,1,\theta}}]$ , i.e. if there is a least of unit of demand belonging to  $D_{q,1,\theta}$  occurring after period  $\tau$ . Thus,  $\tau$  can be used to satisfy part of demand  $\sum_{q \in SD} D_{q,1,\theta}$  only if the resource is setup in  $\tau$  for one of the products such that  $\Delta_{q,D_{q,1,\theta}} \geq \tau$ . This gives  $C_\tau \leq \sum_{q \in SD_\tau} y_{q,\tau}$ .

We thus obtain  $C_\tau \leq \min(\sum_{q \in SD_\tau} y_{q,\tau}, \sum_{p \in SP} y_{pt})$ ,  $\forall \tau \in [1, t-1] \cup [t+1, \theta]$ . It leads to the following inequality stating that, in a feasible solution  $(y, w)$  of the DSLSP, in case period  $t$  is devoted to the production of one product belonging to  $SP$ , the cumulated capacity available for products in  $SD$  over periods  $[1, t-1] \cup [t+1, \theta]$  should be large enough to produce the corresponding cumulated demand:

$$\left[ \sum_{q \in SD} D_{q,1,\theta} \right] \left[ \sum_{p \in SP} y_{pt} \right] \leq \sum_{\substack{\tau=1 \dots t-1 \\ t+1 \dots \theta}} \left[ \min \left( \sum_{q \in SD_\tau} y_{q,\tau}, \sum_{p \in SP} y_{pt} \right) \right] \quad (10)$$

Now, we can exploit our knowledge of the setup state of the resource in period  $t$  to further strengthen this inequality. Namely, we know that a product  $p$  belonging to  $SP$  is produced in period  $t$ . A changeover to (resp. from) this product  $p$  thus has to take place at the beginning (resp. at the end) of period  $t$ . This means that:

- If period  $t-1$  is to be used to satisfy the demand of one of the products belonging to  $SD_{t-1}$ , there must be a changeover from this product  $q \in SD_{t-1}$  to the product  $p \in SP$  at the beginning of period  $t$ . The production capacity available in period  $\tau = t-1$  for the products in  $SD_{t-1}$  is thus limited by  $C_{t-1} \leq \sum_{p \in SP, q \in SD_{t-1}} w_{q,p,t}$ .
- Similarly, if period  $t+1$  is to be used to satisfy the demand of one of the products belonging to  $SD_{t+1}$ , there must be a changeover from the product  $p \in SP$  to this product at the end of period  $t$ . The production capacity available in period  $\tau = t+1$  for the products in  $SD_{t+1}$  is thus limited by  $C_{t+1} \leq \sum_{p \in SP, q \in SD_{t+1}} w_{p,q,t+1}$ .

We can thus strengthen the upper bound of  $C_{t-1}$  (resp  $C_{t+1}$ ) by re-

placing the term  $\min(\sum_{q \in SD_\tau} y_{q,\tau}, \sum_{p \in SP} y_{pt})$  by  $\sum_{p \in SP, q \in SD_{t-1}} w_{q,p,t}$  (resp.  $\sum_{p \in SP, q \in SD_{t+1}} w_{p,q,t+1}$ ) and obtain the inequality (9) discussed in Proposition 1.

This completes the proof.  $\square$

We point out here that, for any integer feasible solution of the DLSPSD, in case  $\sum_{p \in SP} y_{pt} = 1$ , we have:

$$\sum_{j \in SD_\tau} y_{j,\tau} \leq \sum_{p \in SP} y_{pt}, \quad \forall \tau \in [1, t-1] \cup [t+1, \theta] \quad (11)$$

$$\sum_{\substack{p \in SP \\ q \in SD_{t-1}}} w_{q,p,t} = \sum_{q \in SD_{t-1}} y_{q,t-1} \quad \text{if } t \neq 1 \quad (12)$$

$$\sum_{\substack{p \in SP \\ q \in SD_{t+1}}} w_{p,q,t+1} = \sum_{q \in SD_{t+1}} y_{q,t+1} \quad \text{if } t \neq \theta \quad (13)$$

We will thus have  $C_\tau = \sum_{q \in SD} y_{q,\tau}, \forall \tau \in [1, t-1] \cup [t+1, \theta]$  in any integer feasible solution of the problem. However, in a fractional solution obtained by solving the linear relaxation of formulation DLSPSD1, we may encounter situations where  $0 < \sum_{p \in SP} y_{pt} < 1$  so that we may have  $\sum_{p \in SP} y_{pt} \leq \sum_{q \in SD} y_{q,\tau}, \sum_{p \in SP, q \in SD} w_{q,p,t} \leq \sum_{q \in SD} y_{q,t-1}$  and  $\sum_{p \in SP, q \in SD} w_{p,q,t+1} \leq \sum_{q \in SD} y_{q,t+1}$ . In these cases, it is interesting to have the flexibility to select for each period  $\tau$  the smallest upper bound for the available production capacity  $C_\tau$  as this will lead to tighter inequalities.

### 3.2. Illustrative example

We introduce a small instance of the DLSPSD in order to illustrate how the proposed multi-product inequalities may help at strengthening the MILP formulation of the problem. This instance involves  $P = 4$  products and  $T = 10$  periods. Table 1 gives the numerical data on the inventory holding costs, on the changeover costs and on the demand for this instance. We note that the changeover cost matrix displays a frequently encountered feature: the presence of two product families (products  $\{1, 2\}$  and products  $\{3, 4\}$ ). The changeover costs between products belonging to different families are significantly higher than the ones between products belonging to the same family.

We provide in Table 2 the fractional solution obtained by solving the continuous relaxation of formulation DLSPSD1, i.e. the initial formulation of the problem strengthened only by the single-product inequalities (8). In this solution, periods 1 and 2 are assigned to the production of a single product, which complies with the problem constraints. However, in periods 3 to 10, the resource capacity is shared between several products, which means that this solution is not feasible for the integer optimization problem. The corresponding cost,  $Z = 563.25$ , is thus a lower bound for the optimal integer solution value.

The fractional solution provided in Table 2 violates several inequalities belonging to the family described in Proposition 1.

One of them corresponds to the period  $t = 6$ , the interval  $[1; \theta] = [1; 7]$  and the product sets  $SP = \{2\}$  and  $SD = \{3, 4\}$ . Namely, we have:

- $LHS = (D_{3,1,7} + D_{4,1,7})y_{2,6} = 2 * 0.5 = 1$
- $RHS = \sum_{\tau=1}^7 \tilde{C}_\tau = 0.75$  as:
  - $-\tilde{C}_1 = \min(y_{3,1} + y_{4,1}, y_{2,6}) = y_{3,1} + y_{4,1} = 0$       see (9a)
  - $-\tilde{C}_2 = \min(y_{3,2} + y_{4,2}, y_{2,6}) = y_{3,2} + y_{4,2} = 0$       see (9a)
  - $-\tilde{C}_3 = \min(y_{3,3} + y_{4,3}, y_{2,6}) = y_{3,3} + y_{4,3} = 0.25$       see (9a)
  - $-\tilde{C}_4 = \min(y_{3,4} + y_{4,4}, y_{2,6}) = y_{2,6} = 0.5$       see (9a)
  - $-\tilde{C}_5 = w_{3,2,6} + w_{4,2,6} = 0$       see (9b)
  - $-\tilde{C}_6 = 0$       see (9c)
  - $-\tilde{C}_7 = w_{2,3,7} = 0$  (Note that  $SD_7 = \{3\}$ )      see (9d)

We can thus improve the problem formulation by adding the following cut:

$$(D_{3,1,7} + D_{4,1,7})y_{2,6} \leq y_{3,1} + y_{4,1} + y_{3,2} + y_{4,2} + y_{3,3} + y_{4,3} + y_{2,6} + w_{3,2,6} + w_{4,2,6} + w_{2,3,7}$$

The idea underlying this inequality is the following. We choose the subset  $SP = \{2\}$ . If product 2 is not assigned for production in period 6 (i.e.  $y_{26} = 0$ ), the inequality is trivially respected. But if product 2 is assigned for production in period 6 (i.e.  $y_{26} = 1$ ), then we have to make sure that we are able to satisfy the total cumulated demand over the interval  $[1; 7]$  for the products in subset  $SD = \{3, 4\}$  (i.e to satisfy  $D_{3,1,7} + D_{4,1,7}$ ) on the

Table 1: Small illustrative example: numerical input data

$p$	$h_p$	$S_{pq}$					$d_{pt}$									
		$q$					$t$									
		0	1	2	3	4	1	2	3	4	5	6	7	8	9	10
0	0	0	191	156	130	161	0	0	0	0	0	0	0	0	0	0
1	7	152	0	14	122	173	1	0	1	0	1	0	1	0	0	0
2	10	125	13	0	119	157	0	0	0	0	0	0	0	0	1	1
3	6	156	157	109	0	6	0	0	0	0	0	0	1	0	0	1
4	7	116	132	134	19	0	0	0	0	1	0	0	0	0	0	1

remaining periods 1, 2, ..., 5, 7. In this case, the right hand side of inequalities (9) computes a tight upper bound of the production capacity available over these periods for products 3 and 4.

Three additional multi-product nequalities are violated by the fractional solution, namely those corresponding to:

- $t = 4$ ,  $[1; \theta] = [1; 5]$ ,  $SP = \{1\}$  and  $SD = \{4\}$
- $t = 7$ ,  $[1; \theta] = [1; 10]$ ,  $SP = \{3\}$  and  $SD = \{2\}$
- $t = 9$ ,  $[1; \theta] = [1; 10]$ ,  $SP = \{2\}$  and  $SD = \{1, 3, 4\}$

We add these four multi-product inequalities to the formulation DL-SPSD1 and solve the linear relaxation of the resulting strengthened formulation. We obtain the integer feasible solution described in Table 3, the cost of which is  $Z = 574$  and corresponds to the optimal integer solution value.

#### 4. Exact and heuristic algorithms for solving the separation problem

The number of inequalities (9) grows very fast with the problem size. Namely, we have a series of inequalities for the  $\frac{(T+1)T}{2}$  pairs of periods  $(t, \theta)$  with  $t \leq \theta$ . Moreover, for a given pair of periods  $(t, \theta)$ , the number of available inequalities is given by  $3Part(P+1, 3)$  where  $Part(P+1, 3)$  denotes the number of partitions of a set of  $P + 1$  elements into 3 subsets ( $SP$ ,  $SD$  and  $\{0 \dots P\} \setminus (SD \cup SP)$ ).  $Part(P+1, 3)$  can be computed by a mathematical induction on the value of  $P$ . Thus, for an instance involving e.g.  $P = 10$

Table 2: Small illustrative example: solution obtained by computing the continuous relaxation of formulation DLSPSD1

Value of the setup variables  $y_{pt}$

	$t$									
$p$	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0
1	1	1	0.75	0.25	0.5	0	0.5	0	0	0
2	0	0	0	0	0	0.5	0	0.5	0.75	0.25
3	0	0	0	0.5	0	0	0.5	0.25	0.25	0.5
4	0	0	0.25	0.25	0.5	0.5	0	0.25	0	0.25

Table 3: Small illustrative example: solution obtained by computing the continuous relaxation of formulation DLSPSD1 strengthened by 4 multi-product inequalities

Value of the setup variables  $y_{pt}$

	$t$									
$p$	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	1	1
3	0	0	0	0	0	0	1	1	0	0
4	0	0	0	0	1	1	0	0	0	0

products and  $T = 25$  periods, we have  $0.5 * 26 * 25 * 3 * 28501 = 27788475$  inequalities.

The large number of inequalities (9) has two implications. First, it will not be possible to include all of them *a priori* in the MILP formulation of the problem. A cutting-plane generation strategy is thus needed to add to the MILP formulation only a subset of these inequalities. It is described in subsection 4.3. Second, using a simple enumeration scheme to identify which inequalities to incorporate in the formulation will be too time-consuming. Hence the development of efficient algorithms to solve the separation problem is required.

In view of the difficulty of solving the whole separation problem directly, we propose a solution approach relying on its decomposition into a series of smaller separation subproblems, one for each pair of periods  $(t, \theta)$ . Each subproblem thus consists in finding the most violated inequality (9) amongst the ones corresponding to periods  $(t, \theta)$  if such an inequality exists. The resulting combinatorial optimization problem can be solved using either the exact or the heuristic separation algorithms described in subsections 4.1 and 4.2.

#### 4.1. Exact separation algorithm

We first discuss an exact algorithm to solve the separation subproblem encountered for a pair of periods  $(t, \theta)$ . This algorithm is guaranteed to find, amongst the inequalities corresponding to periods  $(t, \theta)$ , the one which is the most violated by the fractional solution  $(\bar{y}, \bar{w})$  if such an inequality exists. The method consists in looking for the partition of  $\{0 \dots P\}$  into 3 subsets providing the largest difference between the left and the right hand sides of inequalities (9). To achieve this, we formulate the separation subproblem for a given  $(t, \theta)$  as a small combinatorial optimization problem.

We first introduce the following decision variables:

- $\alpha_p = 1$  if product  $p$  belongs to subset  $SP$ , 0 otherwise.
- $\beta_p = 1$  if product  $p$  belongs to subset  $SD$ , 0 otherwise.
- $\gamma_\tau = 1$  if capacity  $C_\tau$  is limited by  $\sum_{p=0}^P \bar{y}_{pt} \alpha_p$ , 0 if  $C_\tau$  is limited by  $\sum_{q=0 \dots P \text{ s.t. } \Delta_{q,D_{q,1,\theta}} \geq \tau} \bar{y}_{q\tau} \beta_q$ .

With this notation, the separation problem for a given  $(t, \theta)$  and a solution  $(\bar{y}, \bar{w})$  is formulated as:

$$QBP_{t,\theta}$$

$$\begin{aligned}
max \quad & \sum_{p=0}^P \sum_{q=0}^P D_{q,1,\theta} \alpha_p \beta_q \\
& - \sum_{\substack{\tau=1 \dots t-2 \\ t+2 \dots \theta}} \left[ \sum_{p=0}^P \overline{y_{pt}} \alpha_p \gamma_\tau + \sum_{\substack{q=0 \dots Ps.t. \\ \Delta_{q,D_{q,1,\theta}} \geq \tau}} \overline{y_{q,\tau}} \beta_q (1 - \gamma_\tau) \right] \\
& - \sum_{p=0}^P \sum_{\substack{q=0 \dots Ps.t. \\ \Delta_{q,D_{q,1,\theta}} \geq t-1}} \overline{w_{q,p,t}} \alpha_p \beta_q - \sum_{p=0}^P \sum_{\substack{q=0 \dots Ps.t. \\ \Delta_{q,D_{q,1,\theta}} \geq t+1}} \overline{w_{p,q,t+1}} \alpha_p \beta_q \quad (14) \\
& \alpha_p + \beta_p \leq 1 \quad \forall p \quad (15) \\
& \alpha_p \in \{0, 1\}, \beta_p \in \{0, 1\} \quad \forall p \quad (16) \\
& \gamma_\tau \in \{0, 1\} \quad \forall \tau \quad (17)
\end{aligned}$$

The objective function (14) corresponds to the maximization of the violation of the inequalities, i.e. we seek to identify the subsets  $SP$  and  $SD$  for which the difference between the left and the right hand sides of the inequality takes the largest value. In case the optimal value of (14) is strictly positive, the most violated inequality corresponding to  $(t, \theta)$  has been identified and is described by the optimal values of variables  $\alpha_p$  and  $\beta_p$ . In case this value is less than or equal to 0, it means that no violated inequality exists for the pair of periods  $(t, \theta)$ . Constraints (15) state that a given product  $p$  cannot be simultaneously included in subsets  $SP$  and  $SD$ .

Problem  $QBP_{t,\theta}$  is a binary program with a quadratic objective function and a series of linear constraints. It can be solved to optimality using a quadratic binary programming solver such as the one embedded in CPLEX.

#### 4.2. Heuristic separation algorithm

As can be seen from the computational experiments to be presented in Section 5, solving to optimality a sequence of quadratic binary programs  $QBP_{t,\theta}$  leads to prohibitively long computation times for the cutting-plane generation algorithm, even for small-size instances. We thus propose in what follows a variable depth search heuristic of Kernighan-Lin type to solve the partitioning subproblem  $QBP_{t,\theta}$  for each pair of periods  $(t, \theta)$  (see [12]).

This algorithm relies on a local search procedure: it starts from a feasible partition of the set of products  $\{0, \dots, P\}$  and tries to iteratively improve it



by moving from the current solution to the best solution in the immediate neighbourhood. The neighbourhood of a tripartition  $\Pi$  of  $\{0\dots P\}$  is defined here as the set of tripartitions obtained by moving a single product from its current subset in  $\Pi$  to one of the two other subsets. To avoid cycling between suboptimal solutions and force the exploration of new areas of the search space, a blocking mechanism is used: during a phase of the algorithm, each product can be moved at most once. Once the maximum number of moves has been reached, we stop the algorithm if no improvement has been found during the phase. In case an improvement has occurred, we start a new phase of the algorithm using the best known solution as starting point and unblocking all moves.

A formal statement of the algorithm is as follows:

Start with a tripartition of  $\{0\dots P\}$ ,  $\Pi_{ref}$ , and compute its violation  $V_{ref}$ .  
While (test =0):  
    Let  $test = 1$ ,  $PossMove = P/2$  and  $\Pi_{cur} = \Pi_{ref}$ .  
    Allow all possible moves to explore the neighbourhood of  $\Pi_{cur}$ .  
    While ( $PossMove > 0$ ):  
        Consider the partitions obtained by carrying out all allowed moves in the neighbourhood of  $\Pi_{cur}$  and evaluate the violation for each obtained partition.  
        Select the best partition obtained in this neighbourhood of  $\Pi_{cur}$ ,  $\Pi_{best}$ , forbid the move used to obtain  $\Pi_{best}$  from  $\Pi_{cur}$ , decrease  $PossMove$  by 1 and set  $\Pi_{cur} = \Pi_{best}$ .  
        If  $V_{best} > V_{ref}$ ,  $test = 0$  and  $\Pi_{ref} = \Pi_{best}$ .

In the computational experiments to be presented in Section 5, five distinct partitions are used to initialize the heuristic, namely:

1.  $SP = \emptyset$  and  $SD = \{q = 1\dots P | D_{q,1,\theta} \geq 1\}$
2.  $SP = \{p = 0\dots P | \overline{y_{pt}} > 0\}$  and  $SD = \{q = 1\dots P | q \notin SP \text{ and } D_{q,1,\theta} \geq 1\}$
3.  $SP = \{p_{max}\}$  where  $p_{max} = \arg \max_{p=0\dots P} \overline{y_{pt}}$   
and  $SD = \{q_{max}\}$  where  $q_{max} = \arg \max_{q=1\dots P} \{Need_{q,t}\}$   
with  $Need_{q,t} = D_{q,1,\theta} \overline{y_{p_{max}t}} - w_{p_{max}qt} - w_{q,p_{max},t+1} - \sum_{\tau=1..t-2}^{t+2..t} \min(y_{p_{max},t}, y_{q,\tau})$
4.  $SP = \{p_{max}\}$  where  $p_{max} = \arg \max_{p=0\dots P} \overline{y_{pt}}$   
and  $SD = \{q = 1\dots P | \overline{y_{qt}} > 0 \text{ and } q \notin SP\}$
5.  $SP$  and  $SD$  are randomly generated such that  $SP \cap SD = \emptyset$

In spite of the fact that the above heuristic may not always produce the most violated inequalities, the computational experiments to be presented in Section 5.2 confirm that it consistently produces results of similar quality as compared with the exact separation procedure and this within much smaller computation times.

#### 4.3. Cutting-plane generation algorithm

We now describe the cutting-plane generation algorithm used to strengthen formulation DLSPSD1 by adding a subset of the multi-product inequalities (9).

Several possible strategies for generating cutting-planes could be used. A first one consists in adding, in each iteration, a single inequality (usually the one which is the most violated by the current solution of the linear relaxation) to the formulation. This leads to the introduction of a small number of inequalities and thus limits the increase in the size of the linear program. However, it usually requires a large number of iterations and thus negatively impacts the overall computation time. Another possible strategy is to add, in each iteration, all the violated inequalities found. However, it leads to the introduction of a relatively large number of inequalities and thus to a significant increase in the size of the linear program. This is why an intermediate strategy is exploited in what follows. In each iteration of the cutting-plane generation, we namely seek to identify a single violated inequality for each period  $t$  and add it to the problem formulation. Thus at most  $T$  inequalities are added in each iteration.

Moreover, the inequality added for period  $t$  will not be the most violated one corresponding to period  $t$  but will be amongst the most violated ones. Indeed, to identify the most violated inequality corresponding to period  $t$ , our solution approach would rely on solving a sequence of optimization subproblems  $QBP_{t,\theta}$ , each aiming at finding the most violated inequality for the pair of periods  $(t, \theta)$ , with  $\theta \in [t, T]$ . This amounts to solving the separation problem for period  $t$  by decomposing it into a series of smaller and easier to solve separation subproblems, by solving each subproblem separately and by selecting the most violated inequality amongst the ones found. Thus, finding the most violated inequality for period  $t$  would imply to solve  $T - t + 1$  subproblems, one for each pair of periods  $(t, \theta)$ .

However, our preliminary computational experiments show that when a violated inequality is found for a pair  $(t, \theta)$ , inequalities for pairs  $(t, \theta + 1)$ ,  $(t, \theta + 2)$ ... are most often also violated and the amount of the violation is

identical to the one found for  $(t, \theta)$ . Thus solving a larger amount of separation subproblems does not result in significant improvements in the strength of the inequality found. In other words, it is not computationally efficient to solve all separation subproblems  $(t, \theta)$  for period  $t$ . On the contrary, for each period  $t$ , stopping the search for violated inequalities as soon as one has been found appeared to be a better strategy.

A detailed description of the corresponding cutting-plane generation algorithm is provided below.

```

Compute the initial LP relaxation of the DLSPSD using formulation DLSPSD1.
While (test = 0):
    Denote  $(\bar{y}, \bar{w})$  the solution of the current linear relaxation.
    For  $t = 1 \dots T$  such that  $\exists p$  such that  $0.0001 < \bar{y}_{pt} < 0.9999$ ;
        Let  $\theta = t$  and found = 0.
        While ( $\theta \leq T$  and found = 0),
            Solve the separation problem for periods  $(t, \theta)$  using:
            - either the exact algorithm
            - or the heuristic algorithm as follows:
                Set ini = 1 and foundH = 0
                While (ini < 5 and foundH = 0)
                    - Use the heuristic algorithm with initial partition ini
                    - If a violated inequality is found, foundH = 1
                    - ini = ini + 1
                If a violated inequality has been found, let found = 1.
                 $\theta = \theta + 1$ .
        If a violated inequality has been found during the current iteration for at least one period, add the (at most  $T$ ) violated inequalities found during the current iteration to the problem formulation and compute its LP relaxation.
        Else set test = 1 to stop the cutting-plane generation and solve the problem using a Branch & Bound procedure.

```

## 5. Computational results

We now discuss the results of the computational experiments carried out to evaluate the effectiveness of the proposed multi-product inequalities at

strengthening the formulation of the multi-product DLSPSD and to assess their impact on the total computation time.

### 5.1. Instances

We randomly generated instances of the problem using a procedure similar to the one described in [16] for the DLSP with sequence-dependent change-over costs and times. More precisely, the various instances tested have the following characteristics:

- *Problem dimension.* The problem dimension is represented by the number of products  $P$  and the number of periods  $T$ : we solved medium-size instances involving 4 to 12 products and 10 to 75 periods.
- *Inventory holding costs.* For each product, inventory holding costs have been randomly generated from a discrete uniform  $DU(5, 10)$  distribution.
- *Changeover costs.* We used two different types of structure for the changeover cost matrix  $S$ . Instances of sets A1-A20 have a general cost structure: the cost of a changeover from product  $p$  to product  $q$ ,  $S_{pq}$ , was randomly generated from a discrete uniform  $DU(100, 200)$  distribution. Instances of sets B1-B20 correspond to the frequently encountered case where products can be grouped into product families: there is a high changeover cost between products of different families and a smaller changeover cost between products belonging to the same family. In this case, for products  $p$  and  $q$  belonging to different product families,  $S_{pq}$  was randomly generated from a discrete uniform  $DU(100, 200)$  distribution; for products  $p$  and  $q$  belonging to the same product family,  $S_{pq}$  was randomly generated from a discrete uniform  $DU(0, 100)$  distribution.
- *Production capacity utilization.* Production capacity utilization  $\rho$  is defined as the ratio between the total cumulated demand ( $\sum_{p=1}^P \sum_{t=1}^T d_{pt}$ ) and the total cumulated available capacity ( $T$ ). We set  $\rho = 0.95$  for all instances.
- *Demand pattern.* Binary demands  $d_{pt} \in \{0, 1\}$  for each product have been randomly generated according to the following procedure:

1. We randomly select a product  $p^*$  from a discrete uniform  $DU(1, N)$  distribution and set  $d_{p^*T} = 1$ .
2. For each product  $p$ , except product  $p^*$ , we randomly select a period  $t_p$  from a discrete uniform  $DU(1, T)$  distribution and set  $d_{p,t_p} = 1$ .
3. For each entry in a  $P \times T$  matrix, except for the entries corresponding to the  $(p, t)$  combinations for which we set  $d_{pt} > 0$  in steps 1 or 2, we randomly generate a number  $\alpha_{pt}$  from a discrete uniform  $DU(1, PT)$  distribution.
4. While the total cumulated demand ( $\sum_{p=1}^P \sum_{t=1}^T d_{pt}$ ) does not exceed  $\rho T$ , we consider the entries  $(p, t)$  one by one in the increasing order of the corresponding value  $\alpha_{pt}$  and set  $d_{pt} = 1$ .
5. When the total cumulated demand reaches  $\rho T$ , we examine whether the corresponding instance is feasible by checking that  $\sum_{p=1}^P \sum_{\tau=1}^t d_{p\tau} \leq t$  for all  $t$ . If the instance is infeasible, we repeat steps 1 to 4.

For each considered problem dimension and each possible structure of the changeover cost matrix, we generated 10 instances, leading to a total of 400 instances.

## 5.2. Results

All tests were run on an Intel Core i5 (2.7 GHz) with 4 Go of RAM, running under Windows 7. We used a standard MILP software (CPLEX 12.5) with the solver default settings to solve the problems with one of the following formulations:

- DLPSD1: initial MILP formulation DLSPSD0, i.e. formulation (1)-(7), strengthened by single-product inequalities (8). We used a standard cutting-plane generation strategy based on a complete enumeration of all possible inequalities to add them into the formulation.
- DLSPSD2e: formulation DLSPSD1 strengthened by multi-product inequalities (9). We used the cutting-plane generation algorithm presented in Section 4.3 with the *exact* separation algorithm discussed in Section 4.1.
- DLSPSD2h: formulation DLSPSD1 strengthened by multi-product inequalities (9). We used the cutting-plane generation algorithm presented in Section 4.3 with the *heuristic* separation algorithm discussed in Section 4.2.

Tables 4-6 display the computational results. We provide for each set of 10 instances:

- $P$  and  $T$ : the number of products and planning periods involved in the production planning problem.
- $SP$ : the number of single-product violated inequalities (8) added in the three formulations.
- $MP_e$  and  $MP_h$ : the number of multi-product violated inequalities added in formulation DLSPSD2e by the exact separation algorithm and in formulation DLSPSD2h by the heuristic separation algorithm.
- $G_1$  (resp.  $G_{2e}$ ,  $G_{2h}$ ): the average percentage gap between the linear relaxation of formulation DLSPSD1 (resp. DLSPSD2e, DLSPSD2h) and the value of an optimal integer solution.
- $N_1$  (resp.  $N_{2e}$ ,  $N_{2h}$ ): the average number of nodes explored by the Branch & Bound procedure before a guaranteed optimal integer solution is found or the computation time limit of 2700s is reached.
- $T_1$  (resp.  $T_{2e}$ ,  $T_{2h}$ ): the total computation time (cutting-plane generation and Branch & Bound search) needed to find a guaranteed optimal integer solution (we used the value of 2700s in case a guaranteed optimal integer solution could not be found within the computation time limit).

Table 4 provides the results for the small instances A1-A5 and B1-B5. These results show that the proposed multi-product inequalities (9) are efficient at strengthening the formulation DLSPSD1. Namely, the integrality gap is reduced from an average of 5.8% with formulation DLSPSD1 (see  $G_1$ ) to an average of 1.5% with formulation DLSPSD2e (see  $G_{2e}$ ). We note that this reduction is particularly significant for instances B1-B5 featuring a product family changeover cost structure. Moreover this formulation strengthening is obtained thanks to a relatively small number of multi-product inequalities as can be seen from the average value of  $MP_e$  (14). However, even if the number of nodes needed by the Branch & Bound procedure to find a guaranteed optimal solution is slightly reduced when using formulation DLSPSD2e, it does not lead to an overall reduction of the computation time. This is mainly explained by the fact that the cutting-plane generation algorithm

based on an exact separation algorithm requires prohibitively long computation times to identify the violated multi-product inequalities to be added to the formulation. It is thus necessary to resort to a heuristic separation algorithm such as the one proposed in Section 4.2.

Comparison of the results obtained with the exact and the heuristic separation algorithm for the instances A1-A5 and B1-B5 (see Table 4) shows that the proposed heuristic is very efficient at finding violated inequalities for small size instances. Namely, the average integrality gap for these 100 instances obtained when using the heuristic algorithm is the same as the one obtained when using the exact algorithm (i.e.  $G_{2h} = G_{2e}$ ). Moreover, the average number of violated inequalities found by the heuristic algorithm (13) is nearly the same as the average number of violated inequalities found by the exact algorithm (14).

Results from Tables 5 and 6 confirm that the proposed approach is efficient at strengthening the problem formulation and at reducing the total computation time when the number of periods involved in the planning problem is medium (i.e. below 50). Namely, for instances A5-A15 and B5-B15, the average integrality gap is reduced from 7.6% to 4.1% while using the multi-product inequalities generated with the heuristic separation algorithm. This leads to a significant reduction of the average number of nodes explored by the Branch & Bound algorithm before a proven optimal solution is found (from 2307 to 1201) and consequently to a decrease in the overall average computation time (from 79.3s to 52.2s).

However, when the number of periods increases (i.e. is above 75), the efficiency of the proposed approach at strengthening the MILP formulation and at reducing the computation time seems to decrease. This can be seen by the fact that, for instances A16-A20 and B16-B20, the average number of violated inequalities generated by the cutting-plane generation algorithm decreases as compared to the number of inequalities generated for instances A5-A15 and B5-B15. Consequently, the average integrality gap is only reduced from 7.8% to 7.1% and the average computation time is only decreased from 853.6s to 810.1s. This might indicate that the strength of the proposed multi-product inequalities decreases when the number of planning periods increases.

Table 4: Computational results: small instances

	$P$	$T$	$SP$	DLSPSD1			DLSPSD2e				DLSPSD2h			
				$G_1$	$N_1$	$T_1$	$MP_e$	$G_{2e}$	$N_{2e}$	$T_{2e}$	$MP_h$	$G_{2h}$	$N_{2h}$	$T_{2h}$
A1	4	10	44	0.0%	0	0.1s	0	0.0%	0	0.1s	0	0.0%	0	0.1s
A2	4	15	106	8.7%	1	0.4s	18	0.7%	0	46.5s	16	0.7%	0	0.1s
A3	6	15	108	0.9%	0	0.3s	4	0.2%	0	13.5s	3	0.2%	0	0.1s
A4	4	20	193	2.6%	5	0.4s	15	0.2%	0	445.4s	15	0.2%	0	0.1s
A5	6	20	190	2.3%	4	0.5s	17	0.3%	0	534.4s	13	0.3%	0	0.2s
B1	4	10	41	0.4%	0	0.1s	1	0.0%	0	1.8s	1	0.0%	0	0.1s
B2	4	15	105	11.5%	4	0.3s	15	0.1%	0	23.6s	13	0.1%	0	0.1s
B3	6	15	107	5.3%	1	0.3s	12	2.1%	0	60.1s	11	2.1%	1	0.2s
B4	4	20	192	11.6%	57	0.4s	26	4.4%	7	804.1s	26	4.4%	10	0.3s
B5	6	20	187	14.5%	55	0.8s	38	6.6%	10	1610.7s	35	6.6%	10	0.6s

Table 5: Computational results: instances with a general changeover cost structure

	$P$	$T$	$SP$	DLSPSD1			DLSPSD2h			
				$G_1$	$N_1$	$T_1$	$MP_h$	$G_{2h}$	$N_{2h}$	$T_{2h}$
A6	4	25	295	2.7%	2	0.5s	17	0.3%	0	0.2s
A7	6	25	315	4.3%	9	1.0s	27	0.7%	2	0.5s
A8	8	25	319	2.4%	14	2.5s	44	0.1%	0	0.5s
A9	10	25	318	1.7%	6	1.4s	19	0.0%	0	0.8s
A10	12	25	321	1.9%	14	1.7s	54	0.2%	2	1.7s
A11	4	50	1189	3.8%	60	5.8s	35	1.9%	17	4.7s
A12	6	50	1153	1.6%	32	6.7s	20	0.9%	11	4.0s
A13	8	50	1226	2.9%	142	21.0s	54	1.3%	60	13.3s
A14	10	50	1246	2.1%	99	21.0s	52	1.1%	52	21.7s
A15	12	50	1303	2.0%	147	32.2s	59	0.8%	35	27.3s
A16	4	75	2693	3.4%	483	58.1s	12	3.2%	484	49.8s
A17	6	75	2817	3.5%	1045	129.9s	18	3.3%	1000	132.8s
A18	8	75	2776	2.7%	856	151.9s	23	2.5%	709	130.7s
A19	10	75	2799	1.7%	379	133.7s	18	1.6%	450	148.9s
A20	12	75	2850	2.1%	861	256.8s	39	1.9%	649	224.8s



Table 6: Computational results: instances with a product family changeover cost structure

	P	T	SP	DLSPSD1			DLSPSD2h			
				$G_1$	$N_{IP1}$	$T_I$	$MP_h$	$G_{2h}$	$N_{2h}$	$T_{2h}$
B6	4	25	293	10.2%	38	0.9s	21	4.9%	16	0.5s
B7	6	25	312	9.7%	36	1.3s	31	2.8%	8	0.9s
B8	8	25	320	13.7%	320	5.6s	56	7.1%	95	2.3s
B9	10	25	318	12.7%	309	4.8s	51	4.9%	66	2.9s
B10	12	25	329	11.1%	154	4.6s	96	3.8%	33	4.1s
B11	4	50	1176	11.7%	3327	31.0s	29	8.1%	1637	18.4s
B12	6	50	1248	12.3%	1753	47.7s	48	9.5%	1104	38.8s
B13	8	50	1261	15.1%	8570	232.9s	70	11.7%	5436	200.3s
B14	10	50	1274	15.6%	25837	901.0s	91	11.9%	12707	501.1s
B15	12	50	1271	13.3%	5173	262.6s	142	8.5%	2737	199.8s
B16	4	75	2694	13.4%	20096	635.9s	18	12.2%	18212	574.7s
B17	6	75	2638	11.3%	18895	827.6s	27	10.5%	14416	717.0s
B18	8	75	2688	15.4%	25015	1962.0s	53	13.7%	22501	1815.7s
B19	10	75	2700	12.1%	19087	2085.8s	46	11.2%	16852	2006.9s
B20	12	75	2817	12.8%	16469	2295.2s	70	11.6%	15326	2300.2s

## 6. Conclusion

The multi-product discrete lot-sizing and scheduling problem with sequence-dependent changeover costs has been investigated and a new family of multi-product valid inequalities for the problem has been exhibited. Both an exact and a heuristic separation algorithms have been devised and computationally tested. Our results show that the proposed inequalities are efficient at strengthening the MILP formulation and at reducing the overall computation time needed to obtain guaranteed optimal solution, at least for instances featuring a number of periods up to 50. Among the possible research directions suggested by the present work, it might be worth exploring the extension of the proposed inequalities to other variants of discrete lot-sizing problems, in particular those involving multiple parallel resources or positive changeover times.

## References

- [1] I. Barany, T.J. van Roy and L.A. Wolsey. Strong formulations for multi-item capacitated lot sizing. *Management Science*, vol. 30 (10), pp. 1255-

1261, 1984.

- [2] G. Belvaux and L.A. and Wolsey. Modelling practical lot-sizing problems as mixed-integer programs. *Management Science*, vol. 47(7), pp 993-1007, 2001.
- [3] L. Buschkühl, F. Sahling, S. Helber and H. Tempelmeier. Dynamic capacitated lot-sizing problems: classification and review of solution approaches. *OR Spectrum*, vol. 32, 231-261, 2010.
- [4] S.G. Dastidar and R. Nagi. Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs. *Computers & Operations Research*, vol. 32, pp 2987-3005, 2005.
- [5] G.D. Eppen and R.K Martin. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, vol. 35(6), 832-848, 1987.
- [6] D. Ferreira, A.R. Clark, B. Almada-Lobo and R. Morabito. Single-stage formulations for synchronized two-stage lot-sizing and scheduling in soft drink production. *International Journal of Production Economics*, vol. 136, pp 255-265, 2012.
- [7] B. Fleischmann. The discrete lot sizing and scheduling problem. *European Journal of Operational Research*, vol. 44, pp 337-348, 1990.
- [8] C. Gicquel, M. Minoux and Y. Dallery. On the discrete lot-sizing and scheduling problem with sequence-dependent changeover times. *Operations Research Letters*, vol. 37, pp 32-36, 2009.
- [9] C. Gicquel, M. Minoux and Y. Dallery. Exact solution approaches for the discrete lot-sizing and scheduling problem with identical parallel resources. *International Journal of Production Research*, vol. 49(9), pp 2587-2603, 2011.
- [10] R. Jans and Z. Degraeve. Meta-heuristics for dynamic lot sizing: a review and comparison of solution approaches, *European Journal of Operational Research*, vol. 177, pp 1855-1875, 2007.
- [11] R. Jans and Z. Degraeve. Modelling industrial lot sizing problems: a review. *Industrial Journal of Production Research*, vol. 46(6), pp 1619-1643, 2008.

- [12] B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, vol 49, pp 291307, 1970.
- [13] J. Leung, T.L. Magnanti and R. Vachani. Facets and Algorithms for Capacitated Lot Sizing. *Mathematical Programming*, vol.45, pp. 331-359, 1989.
- [14] A. J. Miller, G. L. Nemhauser, M. W.P. Savelsbergh. On the capacitated lot-sizing and continuous 0-1 knapsack polyhedra. *European Journal of Operational Research*, vol. 125, pp 298-315, 2000.
- [15] Y. Pochet and L.A. Wolsey. Production planning by mixed integer programming, Springer Science, 2006.
- [16] M. Salomon, M. Solomon, L. van Wassenhove, Y. Dumas and S. Dauzère-Pérès. Solving the discrete lotsizing and scheduling problem with sequence dependant set-up costs and set-up times using the Travelling Salesman Problem with time windows. *European Journal of Operational Research*, vol. 100, pp 494-513, 1997.
- [17] C. Silva and J.M. Magalhaes. Heuristic lot size scheduling on unrelated parallel machines with applications in the textile industry. *Computers & Industrial Engineering*, vol. 50, pp 79-89, 2006.
- [18] C.A. van Eijl and C.P.M. van Hoesel. On the discrete lot-sizing and scheduling problem with Wagner-Whitin costs. *Operations Research Letters*, vol. 20, pp 7-13, 1997.
- [19] H.M. Wagner and T.M. Whitin. Dynamic version of the economic lot size model. *Management Science*, vol. 5(1), pp 89-96, 1958.