Tight lower bounds by semidefinite relaxation for the discrete lot-sizing and scheduling problem with sequence-dependent changeover costs

Céline Gicquel, Abdel Lisser, Michel Minoux

9^{th} *International Conference of Modeling, Optimization and Simulation - MOSIM'12*
June 6-8, 2012 - Bordeaux - France
"Performance, interoperability and safety for sustainable development"

TIGHT LOWER BOUNDS BY SEMIDEFINITE RELAXATION FOR THE DISCRETE LOT-SIZING AND SCHEDULING PROBLEM WITH SEQUENCE-DEPENDENT CHANGEOVER COSTS

C. Gicquel, A. Lisser

LRI / Université Paris Sud
Campus d'Orsay, bâtiment 650
91405 Orsay Cedex
celine.gicquel@lri.fr, abdel.lisser@lri.fr

M. Minoux

LIP6 / Université Pierre et Marie Curie
4 place Jussieu
75005 Paris
michel.minoux@lip6.fr

ABSTRACT: *We study a production planning problem known as the discrete lot-sizing and scheduling problem with sequence-dependent changeover costs. This optimization problem can be formulated as a quadratic integer program. In the present paper, we propose to compute a tight lower bound of the optimal solution value by using a semidefinite relaxation of the problem rather than a standard linear relaxation. This is achieved in particular by using reformulation techniques previously proposed in the semidefinite programming literature for the quadratic knapsack problem. The results of the preliminary computational experiments we carried out on small instances show that the proposed approach provides lower bounds of overall improved quality as compared with other possible LP relaxations.*

KEYWORDS: *discrete lot-sizing and scheduling problem, sequence-dependent changerover costs, quadratic integer programming, semidefinite relaxation*

1 INTRODUCTION

Fierce competition in today's global market forces industrial companies to better manage their supply chain networks. In particular, making the right decisions regarding one of the core supply chain processes, goods production, directly affects the productivity and hence the competitiveness of a company. Industrial production management involves, among others, lot-sizing and scheduling decisions.

Lot-sizing arises in production whenever start-up operations such as preheating or tool changing are required in order to prepare the production resource for the processing of a new product. The amount of the related startup costs usually does not depend on the number of products processed after the start-up. Thus, to minimize start-up 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 start-up and inventory holding costs while taking into account both the customer demand satisfaction and the technical limitations of the production system.

We study in the present paper a variant of lot-sizing problem known as the discrete lot-sizing and scheduling problem or DLSP. As defined by (Fleischmann, 1990), in the DLSP, several key assumptions are used to model the production planning problem:
- Demand for products is deterministically known and time-varying.
- A finite time horizon subdivided into discrete periods is used to plan production.
- At most one product can be produced per period and the facility processes either one product at full capacity or is completely idle ("discrete" production policy).
- Costs to be minimized are the inventory holding costs and the changeover costs.

We consider here the complicating case where the changeover costs to be incurred when the production of a new lot begins are sequence-dependent, i.e. depend on both the product produced before and the product produced after the changeover.

A wide variety of solution techniques from the Operations Research field have been proposed to solve

lot-sizing problems: the reader is referred e.g. to the literature reviews provided by (Jans and Degraeve, 2007) and (Buschkühl *et al.*, 2010) for more detail. Among them, most existing exact solution approaches are based on tight mixed-integer linear programming (MILP) formulations which are solved by standard Branch & Bound procedures. In particular, valid inequalities and extended reformulations have been proposed by (Eppen and Martin, 1987) and (van Eijl and van Hoesel, 1997) to obtain tight linear relaxations of the DLSP. However, even if substantial improvements of the lower bounds can be obtained by strengthening the MILP formulation, there are still cases where the linear relaxation of the DLSP is of rather weak quality (see e.g. Gicquel *et al.*, 2009). These difficulties thus motivate the study of more powerful formulations for lot-sizing and scheduling problems. One such possibility consists in using a semidefinite relaxation of the problem rather than the standard continuous relaxation used in mixed-integer linear programming.

To the best of our knowledge, there is no previous attempt at using semidefinite relaxations to solve lot-sizing problems. The main contributions of the present paper are thus twofold. First we propose to compute lower bounds for the DLSP with sequence-dependent changeover costs (DLSPSD) using a semidefinite relaxation rather than a standard linear relaxation. Second we present a cutting-plane generation algorithm based on a semidefinite programming (SDP) solver to tighten the initial semidefinite relaxation. The results of the computational experiments we carried out on small instances show that the proposed approach provides good quality lower bounds on average, especially for the instances featuring a product family cost structure.

The paper is organized as follows. We introduce in Section 2 a quadratic integer programming (QIP) formulation for the DLSPSD. We then explain in Section 3 how this QIP can be reformulated as a semidefinite program and how lower bounds can be obtained for the DLSPSD by semidefinite relaxation. This is achieved mainly by relying on reformulation techniques previously proposed in the SDP literature for the quadratic knapsack problem. Section 4 is devoted to the description of the cutting-plane generation algorithm we use to strengthen the initial semidefinite relaxation of the problem. Some computational results involving a comparison with previously published MILP strengthening techniques are then presented in Section 5.

2 QIP FORMULATION OF THE DLSPSD

We first consider a QIP formulation for the DLSPSD. Namely, the sequence-dependent nature of the changeover costs leads to the introduction of a series of quadratic terms in the objective function.

2.1 Initial QIP formulation

We wish to plan production for a set of products denoted $p = 1...P$ to be processed on a single production machine over a planning horizon involving $t = 1...T$ periods. 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 express the demands to be satisfied as integer numbers of units (see e.g. Fleischmann, 1990 and Gicquel *et al.*, 2011). We denote D_{pt} the demand for product p in period t, 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 a single product to each period of the planning horizon while ensuring demand satisfaction. We thus introduce the binary decision variables y_{pt} where $y_{pt} = 1$ if product p is assigned to period t and 0 otherwise, and obtain the following DLSP1 formulation.

(DLSP1)

$$Z_{DLSP} = min \quad \sum_{p=1}^{P}\sum_{t=1}^{T} h_p \sum_{\tau=1}^{t}(y_{p\tau} - D_{p\tau}) + \sum_{p,q=0}^{P} S_{p,q} \sum_{t=0}^{T-1} y_{pt}y_{qt+1} \quad (1)$$

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

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

$$\forall p, \forall t, \quad y_{pt} \in \{0, 1\} \quad (4)$$

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 and the quadratic term $y_{pt}y_{qt+1}$ is equal to 1 iif the machine is switched from product p to product q at the beginning of period $t+1$. Constraints (2) impose that the cumulated demand over interval $[1, t]$ is sastified by the cumulated production over the same time interval. Constraints (3) ensure that a single product is assigned to each period of the planning horizon.

2.2 Reformulation involving knapsack constraints with positive coefficients

In what follows, we intend to use some of the reformulation techniques proposed for the quadratic knapsack problem (QKP) by (Helmberg, 2000) and (Helmberg *et al.*, 2000) to derive a semidefinite relaxation of the DSLPSD. However, these reformulations require that all coefficients in the knapsack constraint are positive. This is why we carry out a change of decision variables by replacing each binary variable y_{pt} by its complementary variable $x_{pt} = 1 - y_{pt}$.

We thus introduce the binary decision variables x_{pt} where $x_{pt} = 1$ if product p is *not* assigned to period t and 0 otherwise. This leads to the following DLSP2 formulation of the DLSPSD which involves a series of knapsack constraints with positive coefficients.

(DLSP2)

$$Z_{DLSP} = min \sum_{p=1}^{P} \sum_{t=1}^{T} h_p \sum_{\tau=1}^{t} (1 - x_{p\tau} - D_{p\tau}) + \sum_{p,q=0}^{P} S_{p,q} \sum_{t=0}^{T-1} (1 - x_{pt})(1 - x_{qt+1}) \quad (5)$$

$$\forall p, \forall t, \quad \sum_{\tau=1}^{t} x_{p\tau} \leq t - \sum_{\tau=1}^{t} D_{p\tau} \quad (6)$$

$$\forall t, \quad \sum_{p=0}^{P} x_{pt} = P \quad (7)$$

$$\forall p, \forall t, \quad x_{pt} \in \{0, 1\} \quad (8)$$

The objective function (5) corresponds to the minimization of the inventory holding and changeover costs over the planning horizon. Constraints (6) limit the number of non-productive periods for product p over interval $[1, t]$ so as to garantee that there are enough productive periods left to satisfy the cumulated demand for this product over interval $[1, t]$. Constraints (7) ensure that out of the $P+1$ products involved in the production planning problem, exactly P are not assigned to period t of the planning horizon.

3 INITIAL SEMIDEFINITE RELAXATION OF THE DLSPSD

Semidefinite programming can be broadly described as the extension of linear programming from the space of real vectors to the space of symmetric matrices: variables of the optimization problem are semidefinite positive matrices instead of positive real vectors. Semidefinite programming is an area of mathematical programming which has witnessed important developments since the seminal papers of (Lovász and Schrijver, 1991) and (Goemans and Williamson, 1995) were published. A good introduction to the field is provided e.g. by (Helmberg, 2002). In particular, semidefinite relaxations were proved to provide tight bounds for some well-known quadratic combinatorial optimization problems such as the max-cut problem, the quadratic assignment problem or the quadratic knapsack problem.

3.1 Notation and definitions

We denote S_n the set of symmetric matrices of size n. The standard scalar product between two matrices A and B in S_n is defined as:
$< A, B >= tr(AB) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} B_{ij}$
where $tr(.)$ is the trace of a square matrix.

We denote $diag(A)$ the vector containing the main diagonal of a square matrix A and $Diag(a) = a^t.I_n$ the $n \times n$ diagonal matrix containing the vector $a \in \mathbb{R}^n$ on its main diagonal.

A matrix $X \in S_n$ is said to be positive semidefinite iff all its eigenvalues are nonnegative: we write it $X \succeq 0$. The set of positive semidefinite matrices is denoted $S_n^+ \subset S_n$.

A semidefinite program can be defined as the maximization of a linear function of $X \in S_n^+$ subject to a series of linear constraints. Symmetric matrices $C, A_1, ...A_M$ are used to express the objective function and the technical constraints, leading to the following formulation:

$$Z_{SDP} = max < C, X > \quad (9)$$

$$< A_m, X > \leq b_m \qquad \forall m = 1..M \quad (10)$$

$$X \succeq 0 \quad (11)$$

Semidefinite programs are convex optimization problems which can be solved either by interior-point algorithms (see e.g. Benson *et al.*, 2000) or spectral bundle methods (see e.g. Helmberg and Rendl, 2000). They are especially well-suited to deal with quadratic programs. Namely, given $x \in \mathbb{R}^n$, any quadratic form $x^t Ax + b^t x + c$ can be rewritten $< A, xx^t > + b^t x + c$.

3.2 Semidefinite relaxation of the QKP

We now explain how a semidefinite relaxation can be derived from the quadratic binary programming formulation of a standard combinatorial optimization problem: the quadratic knapsack problem. The procedure described in this subsection will serve as a basis for deriving a semidefinite relaxation of the DLSPSD.

The quadratic knapsack problem (QKP) can be for-

mulated as a quadratic binary program where a quadratic objective function is to be maximized under a single linear constraint.

(QKP)

$$Z_{QKP} = max\ z^t \tilde{C} z \quad (12)$$
$$a^t z \leq b \quad (13)$$
$$z \in \{0,1\}^n \quad (14)$$

Note that, since $z_i^2 = z_i$ for $z_i \in \{0,1\}$, linear costs $c^t z$ can be modeled on the diagonal of matrix $\tilde{C}$.

A semidefinite reformulation of (QKP) can be obtained by using the following procedure:
- We introduce the matrix variable $\tilde{Z} = zz^t \in S_n$.
- The objective function is expressed using the scalar product of S_n: $Z_{QKP} =< \tilde{C}, \tilde{Z} >$.
- We introduce a matrix $\tilde{A} = Diag(a) \in S_n$ and express the linear constraint using the scalar product of S_n: $< \tilde{A}, \tilde{Z} >\leq b$.
- The binary character of the decision variables is imposed by stating: $z_i^2 = z_i$ for all i in $[1, n]$, which can be equivalently written as: $diag(\tilde{Z}) = z$

We thus obtain the following SDP formulation:

(QKP)

$$Z_{QKP} = max < \tilde{C}, \tilde{Z} > \quad (15)$$
$$< \tilde{A}, \tilde{Z} >\leq b \quad (16)$$
$$diag(\tilde{Z}) = z \quad (17)$$
$$\tilde{Z} - zz^t = 0 \quad (18)$$
$$\tilde{Z} \in S_n \quad (19)$$

Problem (QKP) cannot be solved as such due to the non-convexity of constraint (18) . We thus replace the matrix equality $Z - zz^t = 0$ by the weaker inequality $Z - zz^t \succeq 0$ (i.e. we enlarge the feasible set by dropping the inequality $Z - zz^t \preceq 0$) and obtain a polynomially solvable convex relaxation of the problem. This provides an initial semidefinite relaxation for the quadratic knapsack problem with $Z_{QKP0} \geq Z_{QKP}$.

(QKP0)

$$Z_{QKP0} = max < \tilde{C}, \tilde{Z} > \quad (20)$$
$$< \tilde{A}, \tilde{Z} >\leq b \quad (21)$$
$$diag(\tilde{Z}) = z \quad (22)$$
$$\tilde{Z} - zz^t \succeq 0 \quad (23)$$
$$\tilde{Z} \in S_n \quad (24)$$

Due to the presence of constraints (22)-(23), (QKP0) cannot be naturally handled by available SDP solvers which are capable of solving semidefinite programs only in the standard form of (9)-(11). This is why we reformulate (QKP0) in the space of symmetric matrices of size $n + 1$.

- Let $Z = \left[\begin{array}{c|c} 1 & z^t \\ \hline z & \tilde{Z} \end{array}\right] \in S_{n+1}$.

Thanks to the Schur complement lemma, we have: $Z \succeq 0$ iif $\tilde{Z} - zz^t \succeq 0$ so that constraint (23) can be replaced with $Z \succeq 0$.

- $C = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \tilde{C} \end{array}\right]$ and $A = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \tilde{A} \end{array}\right]$

are introduced so that the objective function (20) can be reformulated as $Z_{QKP0} = max < C, Z >$ and the linear constraint (21) as $< A, Z >\leq b$.
- Constraint $Z_{11} = 1$ is expressed as $< D_1, Z >= 1$ with $D_1 = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 0 \end{array}\right]$.
- Finally, constraints $diag(Z) = z$ can be expressed as $Z_{ii} - Z_{1i} = 0, \forall i \in [2, n + 1]$. This is enforced in the semidefinite program by constraints $< D_i, Z >= 0$ where $D_i \in S_{n+1}$ is defined as:

$$D_i = \left[\begin{array}{c|ccc} 0 & ... & -0.5 & ... \\ \hline ... & & & \\ -0.5 & ... & 1 & ... \\ ... & & & \end{array}\right]$$

(QKP0) is thus reformulated in the S_{n+1} space in the form suitable for SDP solvers:

(QKP0')

$$Z_{QKP0} = max < C, Z > \quad (25)$$
$$< A, Z >\leq b \quad (26)$$
$$< D_1, Z >= 1 \quad (27)$$
$$< D_i, Z >= 0 \qquad \forall i = 2..n + 1 \quad (28)$$
$$Z \succeq 0 \quad (29)$$
$$Z \in S_{n+1} \quad (30)$$

Now as mentioned in chapter 3 of (Helmberg, 2000), a key ingredient in the design of a semidefinite relaxation for (QKP) is the representation of the linear constraint $a^t z \leq b$ within the quadratic space. (Helmberg, 2000) refers to the reformulation (QKP0) presented above as the "diagonal representation" and explains that a tighter reformulation can possibly be obtained by multiplying both sides of the linear inequality $a^t z \leq b$ by $a^t z$. This can be done only if $a^t z \geq 0$, i.e. if all coefficients of vector a are non negative.

We obtain the inequality $ba^t z - aa^t . zz^t \geq 0$, which can be reformulated in the space S_{n+1} as $< A_1, Z >\geq 0$

with $A_1 = \left[\begin{array}{c|c} 0 & \mathrm{ba}^t/2 \\ \hline \mathrm{ba}/2 & \text{-}\mathrm{aa}^t \end{array}\right]$.

This leads to the formulation of the following semidefinite relaxation of (QKP):

(QKP1)

$$Z_{QKP1} = max < C, Z > \quad (31)$$
$$< A_1, Z > \geq 0 \quad (32)$$
$$< D_1, Z > = 1 \quad (33)$$
$$< D_i, Z > = 0 \qquad \forall i = 2..n+1 \quad (34)$$
$$Z \succeq 0 \quad (35)$$
$$Z \in S_{n+1} \quad (36)$$

(Helmberg, 2000) proves that: $Z_{QKP0} \geq Z_{QKP1} \geq Z_{QKP}$, i.e. that the semidefinite relaxation of (QKP) obtained by using the second reformulation of the linear constraint is as least as strong and possibly tighter than the one obtained by using the diagonal representation. We will thus apply this reformulation technique for all the knaspsack constraints involved in the formulation (DLSP2) of the DLSPSD.

3.3 Semidefinite relaxation of the DLSPSD

Starting from the quadratic integer formulation (DSLP2) of the DLSPSD, we obtain a semidefinite relaxation of the DLSPSD by following a procedure similar to the one described above for the quadratic knapsack problem:
- Let $n = (P+1)T$.
- We define a vector $x = [x_{01}, ..., x_{0T}, ..., x_{pt}, ..., x_{PT}]$ and a matrix $X \in S_{n+1}$ such that $X = \left[\begin{array}{c|c} 1 & \mathrm{x}^t \\ \hline \mathrm{x} & \tilde{X} \end{array}\right]$.
- We define a cost matrix C such that $< C, X >$ is equal to the objective function of formulation (DLSP2).
- Each knapsack linear constraints (6) of the form $a^t x \leq b$ is multiplied by $a^t x$ before being reformulated in the S_{n+1} space. This leads to the introduction of a series of matrices A_{pt} to express the demand satisfaction constraints.
- Each equality linear constraints (7) is reformulated using the diagonal representation. We thus introduce matrices B_t to express the product-period semi-assignment constraints.
- We use matrices D_1 and $D_i, i = 2...n+1$ similar to those used in Section 3.2 to express that $diag(X) = x$.

We thus obtain the following semidefinite program which provides an initial semidefinite relaxation of the DLSPSD.

(SDP0)

$$Z_{SDP0} = < C, X > \quad (37)$$
$$< A_{pt}, X > \geq 0 \qquad \forall p, \forall t \quad (38)$$
$$< B_t, X > = P \qquad \forall t \quad (39)$$
$$< D_1, X > = 1 \quad (40)$$
$$< D_i, X > = 0 \qquad \forall i = 2..n+1 \quad (41)$$
$$X \succeq 0 \quad (42)$$
$$X \in S_{n+1} \quad (43)$$

4 CUTTING-PLANE GENERATION ALGORITHM

4.1 Valid inequalities

We use four families of valid inequalities to strengthen the initial semidefinite relaxation (SDP0) of DLSPSD.

The first family of valid inequalities exploits the presence of implicit binary exclusion constraints implied by the assignment constraints (3). Namely, two different products p and q cannot be assigned to the same period t leading to:

$$\forall t, \forall (p,q) s.t. p \neq q, (1 - x_{pt})(1 - x_{qt}) = 0 \quad (44)$$

We also use two families of valid inequalities discussed in (Helmberg, 2000) for the quadratic knapsack problem and in (Roupin, 2004) for general bivalent quadratic problems. These valid inequalities are obtained by multiplying each knapsack inequality of type (6) either by $x_{qt'}$ or by $(1 - x_{qt'})$. This approach can be seen as a generalization of the reformulation method first proposed by (Sherali and Adams, 1990) to obtain strong relaxations for bivalent linear programs. We thus obtain:

$\forall (p,q)\ \forall (t,t')$,

$$\sum_{\tau=1..t} x_{p\tau} x_{q,t'} \leq (t - \sum_{\tau=1..t} D_{p\tau}) x_{q,t'} \quad (45)$$

$\forall (p,q)\ \forall (t,t')$,

$$\sum_{\tau=1..t} x_{p\tau}(1 - x_{q,t'}) \leq (t - \sum_{\tau=1..t} D_{p\tau})(1 - x_{q,t'}) \quad (46)$$

Finally, we use a fourth family of simple valid inequalities which are part of the so called triangle inequalities used in computing semidefinite relaxation for unconstrained quadratic programs (see e.g. Helmberg,

2000). These are obtained by relying on the fact that we have:

$$\forall(p,q), \forall(t,t'), (1 - x_{pt})(1 - x_{qt'}) \geq 0 \qquad (47)$$

Valid inequalities (44)-(47) are quadratic constraints of the form $\tilde{F}_{ij}x_ix_j + f_ix_i <= g$. We reformulate them by introducing a matrix $F \in S_{n+1}$ such that $F = \left[\begin{array}{c|c} 0 & \mathrm{f}^t/2 \\ \hline \mathrm{f}/2 & \tilde{F} \end{array}\right]$ and add them to formulation SDP0 presented above as $< F, X >\leq g$.

However, the number of valid inequalities (44)-(47) grows very fast with the problem size. It is thus not possible to include all of them directly in formulation SDP0. This is why we devised a cutting-plane generation algorithm to include them as needed in formulation (SDP0).

4.2 Cutting-plane generation algorithm

We use the following cutting-plane generation algorithm (CPA) to strengthen the initial semidefinite relaxation of the DLSPSD obtained by solving formulation SDP0.

Algorithm (CPA)

Step 1
- Solve (SDP0)
- Let test1=test2=test3=test4=0.

Step 2
While $(test1 \times test2 \times test3 \times test4 = 0)$:
Step 2.1
- Look for the p most violated inequalities of type (44) and add them to the current SDP formulation.
- If at least $minCut$ violated inequalities have been found, solve the obtained semidefinite program.
- Else let test1 = 1.

Step 2.2
- Look for the p most violated inequalities of type (45) and add them to the current SDP formulation.
- If at least $minCut$ violated inequalities have been found, solve the obtained semidefinite program.
- Else let test2 = 1.

Step 2.3
- Look for the p most violated inequalities of type (46) and add them to the current SDP formulation.
- If at least $minCut$ violated inequalities have been found, solve the obtained semidefinite program.
- Else let test3 = 1.

Step 2.4
- Look for the p most violated inequalities of type (47) and add them to the current SDP formulation.
- If at least $minCut$ violated inequalities have been found, solve the obtained semidefinite program.
- Else let test4 = 1.

In the numerical experiments presented in Section 5, we used $p = 300$ and $minCut = 5$. Namely, solving large semidefinite programs is very computationally intensive and usually requires a rather large amount of computation time. During the cutting plane generation, we should thus avoid to repeatidly solve semidefinite programs differing from one another only by the addition of a small number of cuts. This is why we try to generate at each step a rather large number of violated cuts (namely $p = 300$) and, in any case, prevent the algorithm from resolving the semidefinite program if less than $minCut = 5$ cuts have been added to the formulation.

When algorithm (CPA) stops, we obtain a lower bound Z_{SDP1} of the optimal integer solution value Z_{DSLP} of problem DSLPSD with $Z_{SDP0} \leq Z_{SDP1} \leq Z_{DSLP}$

5 PRELIMINARY COMPUTATIONAL EXPERIMENTS

We now discuss the results of some preliminary computational experiments carried out to evaluate the quality of the lower bounds provided by the semidefinite relaxation of the DLSPSD discussed in Sections 3 and 4.

5.1 Problem instance generation

We randomly generated instances of the problem using a procedure adapted from that described in (Salomon *et al.*, 1997) for the DLSP with sequence-dependent changeover costs. 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 small-size instances involving 2 to 6 products and 10 to 25 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 A, B, D, F, H, J and L 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 C, E, G, I, K, M 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 ρ 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 $\rho = 0.95$.
- *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 $N \times T$ 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 α_{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 ρT, we consider the entries (p, t) one by one in the increasing order of the corresponding value α_{pt} and set $D_{pt} = 1$.
5. When the total cumulated demand reaches ρT, 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, 10 instances were generated, leading to a total of 130 instances.

5.2 Computational results

For each instance, we compute:
- the lower bound Z_{lp} provided by the linear continuous relaxation of formulation (DLSP1). This linear relaxation is obtained by using the flow-conservation constraints discussed in (Belvaux and Wolsey, 2001) to link the linearization variables with the binary setup variables. It is then further strengthened by the valid inequalities proposed by (van Eijl and van Hoesel, 1997) for the single-item DLSP.
- the lower bound Z_{ext} provided by the extended linear reformulation of (DLSP1). This extended reformulation is adapted from the one proposed by (Eppen and Martin, 1987) for the DLSP with multiple identical parallel production resources.
- the lower bound Z_{sdp1} provided by the semidefinite relaxation of (DLSP2) discussed in Section 3.3 and strengthened by algorithm (CPA),
- the optimal integer solution value Z_{ip} provided by the resolution of the linear reformulation of (DLSPSD) discussed in (Belvaux and Wolsey, 2001).

Linear relaxations are computed using the simplex algorithm embedded in CPLEX 12.1 whereas the optimal integer solution value is obtained using the standard Branch & Bound algorithm embedded in CPLEX 12.1. We use the semidefinite programming solver DSDP based on an interior-point type algorithm (Benson *et al.*, 2000) to solve the various semidefinite programs involved in algorithm (CPA). All tests were run on an Intel Core i5 (2.7 GHz) with 4 Go of RAM, running under Windows 7.

Tables 1 ande 2 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,
- G_{lp} (resp. G_{ext} and G_{sdp}): the average gap between the lower bound Z_{lp} (resp. Z_{ext} and Z_{sdp1}) and the optimal integer solution value Z_{ip}.
- T_{lp} (resp. T_{ext} and T_{sdp}): the average computation time (in seconds) needed to obtain Z_{lp} (resp. Z_{ext} and Z_{sdp1}),
- N_{sdp}: the average total number of cuts generated by algorithm (CPA) to strengthen the SDP relaxation.

Results from tables 1 and 2 show that the lower bounds provided by the semidefinite relaxation of formulation (DLSP2) are of overall improved quality as compared with the strongest linear programming relaxations known for the problem. Namely, for instances with a general changeover cost structure, Z_{sdp1} is slightly tighter than the bounds provided by the best MILP strengthening techniques known for the DLSP. The average gap over the 70 instances is thus decreased from 1.1% with the extended linear reformulation to 0.5% with the semidefinite relaxation. Moreover, for instances with a product family cost structure, the proposed approach provides bounds of significantly better quality than the previously published linear relaxations. This can be seen for instance by the fact than the average gap over the 60 corresponding instances is decreased from 6.2% with the extended linear reformulation to 0.7% with the semidefinite relaxation.

6 CONCLUSION AND PERSPECTIVES

We studied the discrete lot-sizing and scheduling problem with sequence-dependent changeover costs. This optimization problem can be formulated as a quadratic integer progam. We proposed in the

Set	A	B	D	F	H	J	L
P	2	4	4	6	4	6	4
T	10	10	15	15	20	20	25
G_{lp}	2.8	0.0	2.8	0.9	2.6	2.3	2.7
T_{lp}	0.1	0.1	0.1	0.1	0.1	0.1	0.1
G_{ext}	0.8	0.0	1.9	0.3	1.3	2.1	1.4
T_{ext}	0.1	0.1	0.1	0.1	0.2	0.2	0.2
N_{sdp}	93	177	944	977	1769	2246	2296
G_{sdp}	0.2	0.0	0.0	0.1	0.9	1.1	1.1
T_{sdp}	1.7	2.9	116	170	951	1445	1413

Table 1: Results for instances with a general cost structure

Set	C	E	G	I	K	M
P	4	4	6	4	6	4
T	10	15	15	20	20	25
G_{lp}	0.4	11.5	5.3	8.3	8.7	8.3
T_{lp}	0.02	0.1	0.1	0.1	0.1	0.1
G_{ext}	0.0	11.2	4.2	7.2	7.5	7.2
T_{ext}	0.1	0.1	0.1	0.1	0.2	0.2
N_{sdp}	169	598	529	1143	1154	1220
G_{sdp}	0.0	0.0	0.0	1.1	1.9	1.1
T_{sdp}	2.7	163	202	1321	1378	1327

Table 2: Results for instances with a product family cost structure

present paper to compute a lower bound of the optimal integer solution value of the problem by carrying out a semidefinite relaxation. This is achieved mainly by exploiting a reformulation technique proposed by (Helmberg, 2000) for the quadratic knapsack problem. We then presented a cutting-plane generation algorithm relying on four families of non problem-specific valid inequalities to strengthen the initial semidefinite relaxation of the problem. The results of our preliminary computational experiments show that the proposed approach compares well with the best MILP strengthening techniques known for the DLSP. It provides good quality lower bounds on average, especially for the instances featuring a product family cost structure.

However, the computation time needed to obtain the semidefinite lower bounds remains high. This is mainly due to the fact that a series of semidefinite programs of increasing size has to be solved by the cutting-plane generation algorithm. This could be improved thanks to a warm-start strategy: the solution of the previous iteration of the cutting-plane generation algorithm could be used to reoptimize the problem slightly changed by the addition of some valid inequalities. However, as explained in chapter 6 of (Helmberg, 2000), this type of strategy is difficult to implement with interior-point algorithms such as the one embedded in solver DSDP. It might thus be worth investigating the use of a semidefinite programming solver based on a spectral bundle method as this would enable us to more easily exploit a warm-start strategy during the cutting-plane generation.

REFERENCES

Belvaux G. and Wolsey L., 2001. Modelling practical lot-sizing problems as mixed-integer programs. *Management Science*, vol. 47(7), 993-1007.

Benson S.J., Ye Y. and Zhang X., 2000. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, vol. 10(2), 443-461.

Buschkühl L., Sahling F., Helber S. and Tempelmeier H, 2010. Dynamic capacitated lot-sizing problems: classification and review of solution approaches. *OR Spectrum*, vol. 32, 231-261.

Eppen G.D. and Martin R.K, 1987. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, vol. 35(6), 832-848.

Fleischmann B., 1990. The discrete lot sizing and scheduling problem. *European Journal of Operational Research*, vol. 44, 337-348.

Gicquel C., Miègeville N., Minoux M. and Dallery Y., 2009. Discrete lot-sizing and scheduling using product decomposition into attributes. *Computers & Operations Research*, vol. 36, 2690-2698.

Gicquel C., Minoux M. and Dallery Y., 2011. Exact solution approaches for the discrete lot-sizing and scheduling problem with identical parallel resources. *International Journal of Production Research*, vol. 49(9), 2587-2603.

Goemans M.W. and Williamson S.P., 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, vol. 42(6), 1115-1145.

Helmberg C., 2000. Semidefinite programming for combinatorial optimization. Habilitationsschrift, TU Berlin.

Helmberg C., Rendl F., and Weismantel R., 2000. A Semidefinite Programming Approach to the Quadratic Knapsack Problem. *Journal of Combinatorial Optimization*, vol. 4(2), 197-215.

Helmberg C. and Rendl F., 2000. A Spectral Bundle Method for Semidefinite Programming. *SIAM Journal of Optimization*, vol. 10(3), 673-696.

Helmberg C., 2002. Semidefinite programming. *European Journal of Operational Research*, vol. 137(3), 461-482.

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

Lovász B. and Schrijver A., 1991. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, vol. 1, 66-190.

Roupin F., 2004. From linear to semidefinite programming: an algorithm to obtain semidefinite relaxations for bivalent quadratic problems. *Journal of Combinatorial Optimization*, vol. 8, 469-493.

Salomon M., Solomon M., van Wassenhove L., Dumas Y. and Dauzère-Pérès S., 1997. 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. 100,494-513.

Sherali, H.D. and Adams, W.P. A hierarchy of relaxations between continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, vol. 3 (3), 411-430.

van Eijl C.A. and van Hoesel C.P.M., 1997. On the discrete lot-sizing and scheduling problem with Wagner-Whitin costs. *Operations Research Letters*, vol. 20, 7-13.