



HAL
open science

An evaluation of semidefinite programming based approaches for discrete lot-sizing problems

Céline Gicquel, Abdel Lisser, Michel Minoux

► **To cite this version:**

Céline Gicquel, Abdel Lisser, Michel Minoux. An evaluation of semidefinite programming based approaches for discrete lot-sizing problems. *European Journal of Operational Research*, 2014, 237 (2), pp.498-507. 10.1016/j.ejor.2014.02.027 . hal-01120234

HAL Id: hal-01120234

<https://hal.science/hal-01120234>

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.

An evaluation of semidefinite programming based approaches for discrete lot-sizing problems [☆]

C. Gicquel^{a,*}, A. Lisser^a, M. Minoux^b

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

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

Abstract

The present work is intended as a first step towards applying semidefinite programming models and tools to discrete lot-sizing problems including sequence-dependent changeover costs and times. Such problems can be formulated as quadratically constrained quadratic binary programs. We investigate several semidefinite relaxations by combining known reformulation techniques recently proposed for generic quadratic binary problems with problem-specific strengthening procedures developed for lot-sizing problems. Our computational results show that the semidefinite relaxations consistently provide lower bounds of significantly improved quality as compared with those provided by the best previously published linear relaxations. In particular, the gap between the semidefinite relaxation and the optimal integer solution value can be closed for a significant proportion of the small-size instances, thus avoiding to resort to a tree search procedure. The reported computation times are significant. However improvements in SDP technology can still be expected in the future, making SDP based approaches to discrete lot-sizing

[☆]A first version of this paper was presented at the conference MOSIM 2012 (9^e Conférence Internationale de Modélisation, Optimisation et Simulation) in Bordeaux, France.

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

more competitive.

Keywords: manufacturing, discrete lot-sizing and scheduling problem, sequence-dependent changeover costs and times, quadratically constrained quadratic binary programming, semidefinite relaxation, cutting planes

1. Introduction

Capacitated lot-sizing arises in industrial 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 [32]: the authors consider 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 [18].

In the present paper, we focus on one of the variants of lot-sizing problems mentioned in [18], namely the multi-product single-resource discrete lot-sizing and scheduling problem or DLSP. As defined in [9, 18], 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 and/or a changeover time. Changeover costs and times can depend either on the next product only (sequence-independent case) or on the sequence of products (sequence-dependent case). Significant changeover times which consume scarce production capacity tend to further complicate the problem. We consider here the DLSP with sequence-dependent changeover costs and times (denoted DLSPSD in what follows) and assume that the changeover times are expressed as integer numbers of planning periods and satisfy the triangular inequality.

Sequence-dependent changeover costs and times are mentioned in [18] 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 and times have been recently reported in the academic literature: see among others [5] for an injection moulding process, [29] for a textile fibre industry or [8] 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 [4, 17] 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 for 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 efficiency of such a procedure strongly depends on the quality of the lower bounds used to evaluate the nodes of the search tree. Much research has been devoted to the polyhedral study of lot-sizing problems and tight MILP

formulations are now available for many variants of lot-sizing problems: see e.g. [22] for a general overview of the related literature and [2, 7, 10, 31] for contributions focusing specifically on the DLSP.

Nevertheless, even if substantial improvements of the lower bounds can be obtained by using these MILP strengthening techniques, there are still cases where the obtained linear reformulation of the DLSPSD provides lower bounds of rather weak quality (see e.g. the numerical results reported in [10]). These difficulties thus motivate the study of more powerful formulations for the problem. One such possibility consists in using a semidefinite reformulation of the problem rather than the standard linear reformulation used in MILP-based solution approaches.

Semidefinite programming (SDP) is a recent area of mathematical programming which can broadly be 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. Since the seminal papers [12, 20] were published, semidefinite programming and its use to solve quadratic optimization problems have attracted a keen interest among researchers. Thanks to this, there is now a rather good knowledge on how to efficiently reformulate a quadratic optimization problem into a semidefinite program (see e.g. [26]). Semidefinite programming has thus proved succesful at providing tight bounds for some well known quadratic binary problems such as the quadratic knapsack problem or the quadratic assignment problem (see e.g. [13, 24, 34]). However, applications of semidefinite programming in the field of industrial production management are still scarce (see [1, 21, 30] for noticeable exceptions) and to the best of our knowledge, there is no previous attempt at using semidefinite programming to solve lot-sizing problems. The purpose of the present paper is thus to provide a first assessment of the potential of semidefinite programming based approaches to solve discrete lot-sizing problems.

The main contributions of the present paper are thus threefold. First we introduce a new quadratically constrained quadratic binary programming formulation for the DLSPSD. Second, we propose to compute lower bounds for the DLSPSD using a semidefinite reformulation of the problem rather than a standard linear reformulation. Finally we present a cutting-plane generation algorithm based on a semidefinite programming solver to tighten the initial semidefinite relaxation. The results of the computational experiments carried out on small to medium-size instances show that the proposed approach provides lower bounds of significantly improved quality as compared to those

provided by the best previously published linear reformulations, especially for the instances featuring a product family structure. Furthermore, for a high proportion of the small-size instances, the residual gap between the semidefinite relaxation and the optimal integer solution value is entirely closed so that there would be no need to resort to a Branch & Bound procedure to obtain the optimal integer solution. However, due to the limitations of available state-of-the art semidefinite programming solvers, these tight lower bounds are obtained at the expense of significant computation times.

The paper is organized as follows. We introduce in Section 2 a quadratically constrained quadratic binary programming (QCQBP) formulation for the DLSPSD. We then explain in Section 3 how this QCQBP can be reformulated as a semidefinite program and how lower bounds can be obtained for the DLSPSD by semidefinite relaxation. To achieve this, we not only exploit reformulation and strengthening techniques recently proposed in the SDP literature for generic (0-1) quadratic binary problems but also use problem-specific information such as the polyhedral representation of single-product discrete lot-sizing problems. Section 4 is devoted to the description of the valid inequalities used to strengthen the initial semidefinite relaxation of the problem and to the presentation of the cutting-plane generation algorithm implemented to add these valid inequalities iteratively into the initial formulation. Some computational results involving a comparison with the best previously published MILP strengthening techniques are then presented in Section 5.

2. QCQBP formulation of the DLSPSD

We first discuss a new formulation of the DLSPSD as a quadratically constrained quadratic binary (QCQBP) program. The sequence-dependent nature of the changeover costs namely leads to the introduction of a series of quadratic terms in the objective function. Moreover, inequalities involving quadratic terms are needed to ensure that the positive changeover times between different production runs for different products are respected.

2.1. Initial QCQBP 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 $t = 1 \dots T$ 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 apply a preprocessing step on the original demand matrix (see [9, 11]) resulting in a demand matrix containing only 0/1 values. We denote $d_{pt} \in \{0, 1\}$ the demand for product p in period t , h_p the inventory holding cost per unit per period for product p , S_{pq} the sequence-dependent changeover cost to be incurred whenever the resource setup state is changed from product p to product q and Δ_{pq} the required sequence-dependent changeover time between a production run for product p and a production run for 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 respect of required changeover durations.

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 ν_t where $\nu_t = 1$ if the resource is undergoing a transition between production runs at period t and 0 otherwise. This leads to the following DLSPSD1 formulation.

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 $y_{p,t}y_{q,t+\Delta_{pq}+1}$ is equal to 1 if and only if the machine starts a changeover 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 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 undergoing a changeover between two production runs. Constraints (4) and (5) impose that the requested changeover times between production runs are respected. Thus, equalities (4) guarantee that product p can be produced in period t if and only if a changeover from p to another product q (possibly $q = p$) takes place at the beginning of period $t + 1$. Similarly, equalities (5) guarantee that product q can be produced in period t if and only if a changeover from another product p (possibly $p = q$) to product q begins early enough (i.e. in period $t - \Delta_{pq}$) to be finished at the beginning of period t . $\Pi_{p,t}^{start} = \{q = 0 \dots P \text{ s.t. } t + \Delta_{pq} + 1 \leq T\}$ is defined as the set of products towards which it is possible to start a changeover from product p at the beginning of period $t + 1$. Similarly, $\Pi_{q,t}^{end} = \{p = 0 \dots P \text{ s.t. } t - \Delta_{pq} - 1 \geq 0\}$ is the set of products from which a changeover to product q ending at the beginning of period t can be done.

(DLSPSD1)

$$\begin{aligned}
Z_{DLSPSD} = \min & \sum_{p=1}^P \sum_{t=1}^T h_p \sum_{\tau=1}^t (y_{p\tau} - d_{p\tau}) \\
& + \sum_{p,q=0}^P S_{p,q} \sum_{t=0}^{T-\Delta_{pq}-1} y_{pt} y_{q,t+\Delta_{pq}+1}
\end{aligned} \tag{1}$$

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

$$\sum_{p=0}^P y_{pt} + \nu_t = 1, \quad \forall t \tag{3}$$

$$y_{p,t} = \sum_{q \in \Pi_{p,t}^{start}} y_{p,t} y_{q,t+\Delta_{pq}+1} \quad \forall p, \forall t \tag{4}$$

$$y_{q,t} = \sum_{p \in \Pi_{q,t}^{end}} y_{p,t-\Delta_{pq}-1} y_{q,t} \quad \forall q, \forall t \tag{5}$$

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

$$\nu_t \in \{0, 1\} \quad \forall t \tag{7}$$

Formulation (DLSPSD1) can be seen as the quadratic counterpart of the MILP formulation proposed for the DLSPSD in [10]. Namely, introducing the so-called changeover variables defined as $w_{pq,t+1} = y_{p,t} y_{q,t+\Delta_{pq}+1}$ enables us to linearize the proposed QCQBP formulation and leads to the formulation discussed in [10]. We would like to point out that this can only be done if the changeover times satisfy the triangular inequality. Otherwise, we may have situations where $y_{p,t} y_{q,t+\Delta_{pq}+1} = 1$ while no changeover from product p to product q starts at the beginning of period $t+1$ (i.e. $w_{pq,t+1} = 0$). To the best of our knowledge, this is the first time such a QCQBP formulation is proposed for the DLSPSD.

Note that, in case all changeover times are equal to zero, transition variables ν_t are not needed in the formulation, capacity constraints (3) are reformulated as $\sum_{p=0}^P y_{pt} = 1$ and changeover time constraints (4)-(5) are removed from the formulation.

2.2. Reformulation involving knapsack constraints with positive coefficients

In what follows, to derive a semidefinite relaxation of the DSLPSD, we intend to apply on the demand satisfaction constraints a reformulation technique proposed by [15, 19] for linear knapsack constraints. However, this reformulation technique requires that all coefficients in the knapsack constraints are positive. This is why we carry out a change of decision variables by replacing each binary variable y_{pt} by its complementary variable $z_{pt} = 1 - y_{pt}$.

We thus introduce the binary decision variables z_{pt} where $z_{pt} = 1$ if product p is *not* assigned to period t and 0 otherwise. This leads to the following DLSPSD2 formulation of the DLSPSD which involves a series of linear knapsack constraints with positive coefficients to model the customer demand satisfaction.

(DLSPSD2)

$$Z_{DLSPSD} = \min \sum_{p=1}^P \sum_{t=1}^T h_p \sum_{\tau=1}^t (1 - z_{p\tau} - d_{p\tau}) + \sum_{p,q=0}^P S_{p,q} \sum_{t=0}^{T-\Delta_{pq}} (1 - z_{pt})(1 - z_{q,t+\Delta_{pq}+1}) \quad (8)$$

$$\sum_{\tau=1}^t z_{p\tau} \leq t - \sum_{\tau=1}^t d_{p\tau} \quad \forall p, \forall t \quad (9)$$

$$\sum_{p=0}^P z_{pt} - \nu_t = P \quad \forall t \quad (10)$$

$$\sum_{q \in \Pi_{p,t}^{start}} (z_{p,t} z_{q,t+\Delta_{pq}+1} - z_{p,t} - z_{q,t+\Delta_{pq}+1}) = |\Pi_{p,t}^{start}| - 1 - z_{p,t} \quad \forall p, \forall t \quad (11)$$

$$\sum_{p \in \Pi_{q,t}^{end}} (z_{p,t-\Delta_{pq}-1} z_{q,t}) - z_{p,t-\Delta_{pq}-1} - z_{q,t} = |\Pi_{q,t}^{end}| - 1 - z_{q,t} \quad \forall q, \forall t \quad (12)$$

$$z_{pt} \in \{0, 1\} \quad \forall p, \forall t \quad (13)$$

$$\nu_t \in \{0, 1\} \quad \forall t \quad (14)$$

The objective function (8) corresponds to the minimization of the inventory

holding and changeover costs over the planning horizon. Constraints (9) limit the number of non-productive periods for product p over interval $[1, t]$ so as to guarantee that there are enough productive periods left to satisfy the cumulated demand for this product over interval $[1, t]$. Constraints (10) ensure that, in case the resource is undergoing a changeover in period t , then none of the $P + 1$ products involved in the production planning problem can be produced, and that in case there is no changeover during period t , P out of the $P+1$ products are not produced. Quadratic equalities (11)-(12) guarantee that the imposed duration of changeovers between production runs is respected.

3. Initial semidefinite relaxation of the DLSPSD

As mentioned in section 2.1, the current state of the art to solve the DLSPSD is to carry out a linearization of formulation DLSPSD1 by introducing linearization variables w_{pqt} and linking these variables to the variables y_{pt} using a so-called flow formulation of the changeovers (see among others [2, 10]). The obtained linear reformulation can then be strengthened by using the valid inequalities proposed in [10].

In what follows, we investigate another way of solving the problem which does not rely on a linearization of the quadratic formulation but rather uses a semidefinite reformulation.

3.1. Notation and definitions

We first introduce some useful notation and definitions. We refer the reader to the survey provided in [16] for a more comprehensive introduction to the field of semidefinite programming.

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:

$$\langle A, B \rangle = \sum_{i=1}^n \sum_{j=1}^n A_{ij} B_{ij}.$$

I_n denotes the unit matrix of size n , $e \in \mathbb{R}^n$ is the vector of all ones and $e_i \in \mathbb{R}^n$ is the i th unit vector. 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 formed from vector $a \in \mathbb{R}^n$.

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

As explained in [16], semidefinite programming is linear programming over the cone of positive semidefinite matrices S_n^+ . A semidefinite program thus involves a matrix variable $X \in S_n^+$ and deals with the maximization of a linear function of X subject to a series of constraints whose expression are also linear with respect to X . Symmetric matrices C, A_1, \dots, A_M are used to formulate the objective function and the technical constraints. The non-negativity constraints on the vector variable used in linear programming are replaced by semidefiniteness constraints on the matrix variable.

This leads to the following standard formulation for a semidefinite program:

$$Z = \max \langle C, X \rangle \tag{15}$$

$$\langle A_m, X \rangle \leq b_m, \quad \forall m = 1..M \tag{16}$$

$$X \in S_n^+ \tag{17}$$

Semidefinite programs are convex optimization problems which can be solved either by interior-point algorithms (see e.g. [3]) or by spectral bundle methods (see e.g. [14]).

3.2. Semidefinite relaxation of the DLSPSD

We now explain how a semidefinite relaxation can be derived from the QC-QBP formulation of the DLSPSD. To carry out this reformulation, we rely on reformulation techniques recently developed for generic 0-1 quadratic programs (see e.g. [15, 20, 23]). Moreover, as pointed out by several authors ([13, 26]), semidefinite relaxations of significantly improved quality can be obtained by applying some specific pretreatments to the linear (equality or inequality) constraints of the original QCQBP: in what follows, we exploit this knowledge to improve the quality of the bounds provided by the semidefinite relaxation of the DLSPSD. One of these pretreatments requires that linear knapsack inequality constraints involve only non-negative coefficients: this is why we start from formulation DLSPSD2 (instead of formulation DLSPSD1) to derive the semidefinite relaxation of the DLSPSD.

3.2.1. Reformulation in the space S_{n+1}

The first step of the reformulation consists in lifting the problem from the space of real vectors to the higher dimensional space of the symmetric matrices.

Let $n = (P + 2)T$.

We first define the vector $x \in \mathbb{R}^n$:

$$x = [z_{01}, \dots, z_{0T}, \dots, z_{pt}, \dots, z_{pT}, \nu_1, \dots, \nu_T]^t = [x_i]_{i=1 \dots n}$$

and introduce the following matrix $X \in S_{n+1}$.

$$X = \left[\begin{array}{c|cc} 1 & & x^T \\ \hline x & & xx^T \end{array} \right] = \left[\begin{array}{c|cccc} 1 & x_1 & x_2 & \dots & x_n \\ \hline x_1 & x_1^2 & x_1x_2 & \dots & x_1x_n \\ x_1 & x_1x_2 & x_2^2 & \dots & x_1x_2 \\ \vdots & & & & \\ x_n & x_1x_n & x_1x_n & \dots & x_n^2 \end{array} \right] \in S_{n+1}$$

We would like to point out here that each possible quadratic term $x_i x_j$ now corresponds to a coefficient X_{ij} of matrix X . Thus, given a quadratic expression involving vector x such as $x^t \tilde{Q} x + q^t x$, we can obtain an equivalent linear expression involving matrix X , i.e. $x^t \tilde{Q} x + q^t x = \langle Q, X \rangle$ where $Q \in S_{n+1}$ is defined as:

$$Q = \left[\begin{array}{c|c} 0 & q^T/2 \\ \hline q/2 & \tilde{Q} \end{array} \right]$$

The second step of the reformulation consists in reformulating the initial QCQP formulation as a linear optimization problem in space S_{n+1} .

The reformulation of the objective function (8) is straightforward. Namely, (8) is a quadratic expression of the form $x^t \tilde{C} x + c^t x$ and can therefore be reformulated as $\min \langle C, X \rangle$ where C is defined using matrix \tilde{C} and vector c as explained above.

We now consider the reformulation of the customer demand satisfaction constraints (9). These constraints can be seen as linear knapsack constraints of the form $a^t x \leq b$ with $a_i \geq 0, \forall i \in [1, n]$. A straightforward way of reformulating each of these linear inequalities would be to use a so-called "diagonal representation" (see [13]): we add a zero coefficient to vector a to obtain a vector $[0, a] \in \mathbb{R}^{n+1}$, introduce the diagonal matrix $Diag([0, a]) \in S_{n+1}$ and reformulate the knapsack constraint as: $\langle Diag([0, a]), X \rangle \leq b$. However a better semidefinite relaxation can be obtained by applying a pretreatment of these constraints before reformulating them in the SDP. This pretreatment consists in multiplying both sides of the linear inequality $a^t x \leq b$ by $a^t x$ (see [15, 19, 26]). This can be done only if $a^t x \geq 0, \forall x \geq 0$, i.e. if all

coefficients of vector a are non negative. We obtain the quadratic inequality $-x^t a a^t x + b a^t x \geq 0$, which can be reformulated $\langle A, X \rangle \geq 0$ where A is defined using matrix $a a^t$ and vector $b a^t$ as explained above. We carry out this reformulation on each of the PT customer demand satisfaction constraints (9), which leads to a series of constraints: $\langle \tilde{A}_{pt}, \tilde{X} \rangle \geq 0 \quad \forall p, \forall t$ in the semidefinite reformulation of DLSPSD2.

We then deal with the reformulation of the resource capacity constraints (10). These are linear equality constraints of the form $a^t x = b$. Following the recommendations found in [23, 26], we first reformulate each of them using the "diagonal representation" $\langle \text{Diag}([0, a]), X \rangle = b$. We then seek to improve the reformulation through the use of a "square representation" of these linear equalities. This is done by squaring both sides of the equality. This leads to quadratic equalities of the form $x^T a a^t x = b^2$ which can then be reformulated in space S_{n+1} . We thus obtain $2T$ equality constraints of the form: $\langle B_t^d, X \rangle = P, \quad \forall t$ and $\langle B_t^s, X \rangle = P^2, \quad \forall t$.

Constraints (11)-(12) ensuring that changeover times between production runs for different products are respected are quadratic equalities of the form $x^t \tilde{F} x + f^t x = \theta$ and can therefore be reformulated as $\langle F, X \rangle = \theta$ where matrix F is obtained from matrix \tilde{F} and vector f as explained above. We carry out this reformulation on each of the $2PT$ changeover times satisfaction constraints, which leads to a series of constraints: $\langle F_{p,t}^{start}, X \rangle = |\Pi_{p,t}^{start}| - 1 \quad \forall p, \forall t$ and $\langle F_{q,t}^{end}, X \rangle = |\Pi_{q,t}^{end}| - 1 \quad \forall q, \forall t$ in the semidefinite reformulation of DLSPSD2.

We finally focus on the constraints (13)-(14) imposing the binary character of the decision variables. We note that $x \in \{0, 1\}^n$ is equivalent to $x_i^2 = x_i, \forall i \in [1, n]$, i.e. to $X_{ii} - 0.5X_{i0} - 0.5X_{0i} = 0, \quad \forall i \in [1, n]$. This is enforced in the semidefinite reformulation by a series of constraints of the form $\langle D_i, X \rangle = 0$

where $D_i \in S_{n+1}$ is defined as:
$$D_i = \left[\begin{array}{c|c} 0 & e_i^t/2 \\ \hline e_i/2 & e_i e_i^t \end{array} \right].$$

We thus obtain the following semidefinite reformulation of formulation DLSP2 in the space S_{n+1} .

(SDP)

$$Z_{DLDP\text{SD}} = \min \langle C, X \rangle \quad (18)$$

$$\langle A_{pt}, X \rangle \geq 0, \quad \forall t, \forall p \quad (19)$$

$$\langle B_t^d, X \rangle = P, \quad \forall t \quad (20)$$

$$\langle B_t^s, X \rangle = P^2, \quad \forall t \quad (21)$$

$$\langle F_{p,t}^{\text{start}}, X \rangle = |\Pi_{p,t}^{\text{start}}| - 1 \quad \forall p, \forall t \quad (22)$$

$$\langle F_{q,t}^{\text{end}}, X \rangle = |\Pi_{q,t}^{\text{end}}| - 1 \quad \forall q, \forall t \quad (23)$$

$$\langle D_i, X \rangle = 0, \quad \forall i \in [1, n] \quad (24)$$

$$X = \left[\begin{array}{c|c} 1 & x^T \\ \hline x & xx^T \end{array} \right] \quad (25)$$

In case all changeover times are equal to zero, the proposed reformulation of the problem can be adapted in a straightforward manner. As transition variables ν_t are not introduced into the formulation, we define $n = (P + 1)T$ and $x = [z_{01}, \dots, z_{0T}, \dots, z_{pt}, \dots, z_{PT}]$. Constraints (22)-(23) are not included in the formulation, the other constraints are reformulated as described above.

3.2.2. Convex relaxation

Problem (18)-(25) is equivalent to the initial QCQP problem. However it cannot be solved as such due to the presence of the nonconvex constraint (25). We thus carry out a convex relaxation, i.e. we enlarge the feasible set of the problem to make it convex by dropping some of the constraints of the problem.

This convex relaxation can be explained as follows (see e.g. [13]).

We first note that:

$$X = \left[\begin{array}{c|c} 1 & x^T \\ \hline x & xx^T \end{array} \right] \Leftrightarrow \begin{cases} X_{00} = 1 \\ X \succeq 0 \\ \text{rank}(X) = 1 \end{cases} \Leftrightarrow \begin{cases} \langle D_0, X \rangle = 1 \\ X \succeq 0 \\ \text{rank}(X) = 1 \end{cases}$$

where D_0 is a $(n + 1) \times (n + 1)$ matrix where all coefficients are 0 except $D_0^{00} = 1$.

We then relax the problem by dropping the rank one constraint. This leads to the formulation of a standard semidefinite program such as (15)-(17) which is a convex optimization problem solvable by available SDP solvers. However, as some constraints of the original problem have been removed, the corresponding optimal solution value will only provide a lower bound on the integer optimal solution value Z_{DLSPSD} . We denote it $Z_{SDP0} \leq Z_{DLSPSD}$. We thus obtain the following semidefinite program which provides an initial semidefinite relaxation of the DLSPSD.

(SDP0)

$$Z_{SDP0} = \min \langle C, X \rangle \quad (26)$$

$$\langle A_{pt}, X \rangle \geq 0, \quad \forall t, \forall p \quad (27)$$

$$\langle B_t^d, X \rangle = P, \quad \forall t \quad (28)$$

$$\langle B_t^s, X \rangle = P^2, \quad \forall t \quad (29)$$

$$\langle F_{p,t}^{start}, X \rangle = |\Pi_{p,t}^{start}| - 1 \quad \forall p, \forall t \quad (30)$$

$$\langle F_{q,t}^{end}, X \rangle = |\Pi_{q,t}^{end}| - 1 \quad \forall q, \forall t \quad (31)$$

$$\langle D_i, X \rangle = 0, \quad \forall i \in [1, n] \quad (32)$$

$$\langle D_0, X \rangle = 1 \quad (33)$$

$$X \succeq 0 \quad (34)$$

4. Strengthening of the initial semidefinite relaxation

4.1. Valid inequalities

As explained above, solving formulation SDP0 provides an initial lower bound Z_{SDP0} of the optimal integer value Z_{DLSPSD} of the DLSPSD. This lower bound can be improved by strengthening the initial semidefinite relaxation of the DLSPSD. In what follows, we propose to achieve this by using five families of valid inequalities: one family exploiting some specific features of the problem under study and four families which have been proposed to strengthen semidefinite relaxations of generic quadratic binary problems.

4.1.1. Problem-specific valid inequalities

We first consider a family of valid inequalities proposed for the single-product DLSP with sequence-dependent changeover times in [10]. These valid inequalities can be seen as an extension of the valid inequalities developed

in [31] for the case without changeover times. They are shown in [10] to be rather efficient at strengthening the continuous relaxation of the linear reformulation of the DLSPSD.

We introduce some additional notation to express these constraints:

- $d_{p,t,\tau}$: cumulated demand for product p in the interval $\{t, \dots, \tau\}$.
- $\theta_{p,v}$: period where the v^{th} unit demand for product p occurs. Note that $\theta_{p,d_{p,1,t+v}}$ denotes the period in which the v^{th} positive demand for product p after period t occurs.

The following constraints are valid inequalities for the DLSPSD.

$$\begin{aligned}
& \sum_{\tau=1}^t (1 - z_{p\tau}) - \sum_{\tau=1}^t d_{p\tau} \\
& + \sum_{v=1}^w \left[(1 - z_{p,t+v}) + \sum_{\tau=t+v+1}^{\theta_{p,d_{p,1,t+v}}} \sum_{q \in \Pi_{p,\tau}^{\text{end}}, q \neq p} (1 - z_{q,\tau - \Delta_{qp} - 1}) (1 - z_{p,\tau}) \right] \\
& \geq w \\
& \forall p, \forall t, \forall w \in [1, d_{p,t+1,T}] \quad (35)
\end{aligned}$$

The idea underlying (35) is to make sure, that, at the end of period t , we will be capable of satisfying the forthcoming w unit demands for product p , either by relying on units of product p currently in inventory or by producing them on the resource within the time interval $[t+1, \theta_{p,d_{p,1,t+w}}]$. The first term of the left hand side of inequalities (35) thus computes the inventory level of product p at the end of period t as the difference between the cumulated production and the cumulated demand for this product over the interval $[1, t]$. The second term computes the production capacity available for product p within interval $[t+1, \theta_{p,d_{p,1,t+w}}]$ by considering the resource setup states on these periods. Inequalities (35) thus state that the sum of these two terms (inventory + production capacity) should be large enough to satisfy the first w units of demand for product p occurring after period t .

We now explain in more detail how the production capacity available for product p within interval $[t+1, \theta_{p,d_{p,1,t+w}}]$ is computed. We first consider the case where $w = 1$, i.e. the case where we focus on satisfying the first unit of demand for product p after period t . If the unit demand is to be produced in

the interval $[t + 1; \theta_{p,d_{p,1,t+1}}]$, the resource either has to be setup for product p in period $t + 1$ (in which case the term $1 - z_{p,t+1}$ is equal to one) or has to undergo a changeover from a product q to product p to allow production of product p in at least one of the periods $t + 2 \dots \theta_{p,d_{p,1,t+1}}$ (in which case at least one of the terms $(1 - z_{q,\tau - \Delta_{qp} - 1})(1 - z_{p,\tau})$ is equal to one). This reasoning can be generalized by a mathematical induction on w to take into account not only the first unit demand for product p after period t ($w = 1$), but any possible number of unit demands ($w \in [1, d_{p,t+1,T}]$).

4.1.2. Generic valid inequalities

We now explain how some of the valid inequalities recently proposed for strengthening the semidefinite relaxation of generic quadratic binary programs can be used for computing tight lower bounds for the DLSPSD.

The first family of generic valid inequalities exploits the presence of implicit binary exclusion constraints implied by the resource capacity constraints (3) and changeover time satisfaction constraints (11)-(12). Namely, assigning a product p to a given period t is not compatible with assigning another product q to any period $\tau \in [t; t + \Delta_{pq}]$. This leads to the following valid equalities:

$$(1 - z_{pt})(1 - z_{q\tau}) = 0 \quad \forall p, \forall q \text{ s.t. } p \neq q, \forall t, \forall \tau \in [t; t + \Delta_{pq}] \quad (36)$$

We also use two families of valid inequalities discussed among others in [13] for the quadratic knapsack problem and in [26] for general bivalent quadratic problems. These valid inequalities are obtained by multiplying each knapsack inequality of type (9) either by $z_{qt'}$ or by $(1 - z_{qt'})$. This approach can be seen as a generalization of the reformulation method first proposed by [28] to obtain strong relaxations for bivalent linear programs. We thus obtain:

$$\sum_{\tau=1..t} z_{p\tau} z_{q,t'} \leq (t - \sum_{\tau=1..t} d_{p\tau}) z_{q,t'}, \quad \forall p, \forall q, \forall t, \forall t' \quad (37)$$

$$\sum_{\tau=1..t} z_{p\tau} (1 - z_{q,t'}) \leq (t - \sum_{\tau=1..t} d_{p\tau}) (1 - z_{q,t'}), \quad \forall p, \forall q, \forall t, \forall t' \quad (38)$$

We note here that a similar strengthening technique could have been applied on the linear equality constraints (10). However, the preliminary computational experiments we carried out showed that it neither lead to better lower bounds nor improve the overall computation time. This is why we do not include them in our strengthening procedure.

Finally, we use a family of simple valid inequalities which are part of the triangle inequalities used in computing semidefinite relaxation for unconstrained quadratic programs (see e.g. [13]). These are obtained by relying on the fact that we have:

$$(1 - z_{pt})(1 - z_{qt'}) \geq 0, \quad \forall p, \forall q, \forall t, \forall t' \quad (39)$$

Valid inequalities (35)-(39) are quadratic constraints of the form $\tilde{x}^t \tilde{F} x + f^t s \leq g$. We reformulate them in the semidefinite program by introducing a matrix

$$F \in S_{n+1} \text{ such that } F = \left[\begin{array}{c|c} 0 & f^t/2 \\ \hline f/2 & \tilde{F} \end{array} \right] \text{ and add them to formulation SDP0}$$

as $\langle F, X \rangle \leq g$.

The number of valid inequalities (35)-(39) grows very fast with the problem size. It is therefore not possible to include all of them directly in the initial formulation SDP0. A cutting-plane generation algorithm is thus proposed in the next subsection to iteratively include a subset of these valid inequalities 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

- Define the initial formulation (SDP0)
- Add all valid inequalities of type (35).
- Add all valid inequalities of type (39) corresponding to $t' = t + \Delta_{pq} + 1$.
- Solve the resulting strengthened formulation (SDP1).
- Let $test=0$.

Step 2

While ($test = 0$):

- Consider the optimal solution of the current semidefinite program.
- Remove all added inequalities inactive in the current solution, i.e. having a strictly positive slack variable.
- Look up for the p_1 most violated inequalities of type (35).
- Look up for the p_2 most violated equalities of type (36).
- Look up for the p_3 most violated inequalities of type (37).
- Look up for the p_4 most violated inequalities of type (38).
- Look up for the p_5 most violated inequalities of type (39).
- Add the selected violated inequalities of each family to the current semidefinite program.
- If at least $minCuts$ violated inequalities have been found during the current iteration, solve the strengthened semidefinite program (SDP2).
- Else set $test = 1$ to stop the algorithm.

Step 1 of algorithm (CPA) starts with the initial semidefinite relaxation (SDP0) of the problem. Our computational experiments showed that this initial formulation provides rather poor lower bounds. For some instances, we even have $Z_{sdp0} \leq 0$ which means that the gap between the lower bound provided by the semidefinite relaxation and the integer optimal solution value is larger than 100%. This is due among others to the fact that the coefficients of the variable matrix X violate a large proportion of the triangle inequalities (39). In particular, they do not comply with the subset of the triangle inequalities (39) for which $t' = t + \Delta_{pq} + 1$: $(1 - z_{pt})(1 - z_{q,t+\Delta_{pq}+1}) \geq 0$. The left hand side of these inequalities corresponds to a term with a positive cost coefficient S_{pq} in the objective function (8) of the problem. Thus, in case $X_{pT+t,qT+t+\Delta_{pq}+1} - X_{1,pT+t} - X_{1,qT+t+\Delta_{pq}+1} + 1 < 0$ in the current SDP solution, the corresponding term in the objective value takes a negative value, which decreases the quality of the obtained lower bound. This is why we add *a priori* to the initial formulation of the problem all valid

inequalities of family (39) corresponding to $t' = t + \Delta_{pq} + 1$ to improve the quality of the initial lower bound Z_{sdp0} and reduce the number of iterations of the cutting-plane generation algorithm. Similarly, we add *a priori* all single-product valid inequalities (35) as this leads to an overall decrease in the number of cutting-plane generation iterations needed to strengthen the semidefinite relaxation.

At the end of step 1 of algorithm (CPA), we obtain a lower bound Z_{SDP1} of the optimal integer solution value Z_{DSLPSD} of problem DSLPSD with $Z_{SDP0} \leq Z_{SDP1} \leq Z_{DSLPSD}$.

Step 2 of algorithm (CPA) starts with removing the inequalities added in the formulation which are inactive for the current primal solution. Theoretically, the redundancy of a primal constraint should be indicated by a zero dual variable. However, as mentioned e.g. in [6], interior-point algorithms terminate at approximate solutions that are still interior so that the complementary condition is never satisfied exactly and dual variables of redundant inequalities might take values significantly different from zero. This is why we choose to remove valid inequalities from the formulation using only the value of their primal slack variable.

Step 2 of algorithm (CPA) then goes on with the addition of valid inequalities which are violated by the current solution. In the numerical experiments presented in Section 5, we used $p_1 = p_2 = p_3 = p_4 = 100$, $p_5 = 300$ and $minCuts = 200$. Namely, solving large semidefinite programs is computationally intensive and usually requires a rather large amount of computation time. During the cutting plane generation, we should thus avoid to repeatedly 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 belonging to the different families of valid inequalities and, in any case, we prevent the algorithm from resolving the semidefinite program if less than $minCuts$ cuts have been added to the formulation.

When algorithm (CPA) stops, we obtain a lower bound Z_{SDP2} of the optimal integer solution value Z_{DSLPSD} of problem DSLPSD with $Z_{SDP0} \leq Z_{SDP1} \leq Z_{SDP2} \leq Z_{DSLPSD}$.

5. Computational experiments

We now discuss the results of some 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 similar to the one described in [27] for the DLSP with sequence-dependent changeover 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 small to medium-size instances involving 4 to 6 products and 15 to 50 periods.
- *Changeover times.* We have experimented with problem instances without changeover times (sets A1-A11 and A1f-A11f) and problem instances with changeover times (sets B1-B9 and B1f-B9f). Among the instances with positive changeover times, instances of sets B1-B9 have a general changeover time structure where the changeover times are randomly generated from a discrete uniform $DU(0, 1)$ distribution. Instances B1f-B9f correspond to the case where products can be grouped into families. In this case, there is a changeover time of 1 period between products belonging to different families and no changeover time between products belonging to the same family.
- *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-A11 and B1-B9 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 A1f-A11f and B1f-B9f 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 $\rho = 0.95$ for instances with zero changeover times and $\rho = 0.80$ for instances with positive changeover times.
- *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 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.

We generated 220 instances with zero changeover times and 180 instances with positive changeover times, leading to a total of 400 instances.

5.2. Computational results

For each instance, we compute:

- the lower bound Z_{lp} provided by the linear relaxation of DLSPSD1 discussed in [10] and strengthened by the valid inequalities proposed by [10] for the single-product DLSP.
- the lower bound Z_{sdp1} provided by the semidefinite relaxation of DLSPSD2 discussed in Section 3 and strengthened by step 1 of algorithm (CPA).

- the lower bound Z_{sdp2} provided by the semidefinite relaxation of DLSPSD2 discussed in Section 3 and strengthened by steps 1 and 2 of algorithm (CPA).
- the optimal integer solution value Z_{ip} obtained by applying on the problem a Branch & Bound procedure using the lower bound Z_{lp} at each node of the search tree.

Linear programs are computed using the simplex algorithm embedded in CPLEX 12.5 whereas the optimal integer solution value is obtained using the standard Branch & Bound algorithm embedded in CPLEX 12.5. We use the semidefinite programming solver DSDP based on an interior-point type algorithm (see [3]) to solve the various semidefinite programs involved in algorithm (CPA). All tests were run on an Intel Core i5 (2.7 GHz) with 4 GB of RAM, running under Windows 7.

Tables 1-4 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.
- V_{lp} and C_{lp} : the number of variables and constraints in the linear relaxation.
- $n + 1$ and C_{sdp} : the size of the variable matrix X and the number of constraints involved in the initial semidefinite relaxation.
- Cut_1 : the average number of cuts added to the SDP formulation by step 1 of algorithm (CPA) and Cut_2 the average total number of cuts generated by steps 1 and 2 of algorithm (CPA).
- G_{lp} (resp. G_{sdp1} and G_{sdp2}): the average percentage gap between the lower bound Z_{lp} (resp. Z_{sdp1} and Z_{sdp2}) and the optimal integer solution value Z_{ip} .
- Opt : the number of instances for which the proposed semidefinite programming based approach provides the exact optimal solution value of the DLSPSD.
- T_{lp} (resp. T_{sdp1} and T_{sdp2}): the average computation time (in seconds) needed to obtain Z_{lp} (resp. Z_{sdp1} and Z_{sdp2}).

Table 1: Results for instances without changeover times: general cost structure

Set	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
P	4	6	4	6	4	6	4	6	4	6	4
T	15	15	20	20	25	25	30	30	40	40	50
V_{lp}	450	815	600	1120	750	1375	870	1650	1160	2200	1450
C_{lp}	353	453	530	647	653	881	875	1041	1299	1607	1189
G_{lp}	2.8%	0.9%	2.6%	2.3%	2.7%	4.3%	1.5%	1.6%	2.7%	1.3%	3.8%
T_{lp}	0.1s	0.1s	0.1s	0.1s	0.1s	0.1s	0.2s	0.3s	0.3s	0.4s	0.4s
$n+1$	76	106	101	141	126	176	151	211	201	281	251
C_{sdp}	124	157	164	206	204	256	243	305	323	405	403
C'_{ut_1}	408	726	610	1025	833	1364	1075	1695	1634	2486	2334
G_{sdp1}	1.4%	0.6%	2.0%	1.5%	2.2%	3.1%	1.0%	1.1%	2.5%	1.2%	3.5%
T_{sdp1}	8s	20s	20s	52s	449s	120s	86s	208s	285s	664s	819s
C'_{ut_2}	614	854	932	1496	1337	2238	1251	2042	2202	2736	2563
G_{sdp2}	0.0%	0.0%	0.0%	0.01%	0.0%	0.1%	0.3%	0.2%	1.1%	0.6%	2.2%
Opt_2	10	10	10	9	10	8	8	7	4	3	1
T_{sdp2}	26s	55s	89s	247s	219s	989s	221s	592s	1217s	2252s	2669s

Table 2: Results for instances without changeover times: product family cost structure

Set	A1f	A2f	A3f	A4f	A5f	A6f	A7f	A8f	A9f	A10f	A11f
P	4	6	4	6	4	6	4	6	4	6	4
T	15	15	20	20	25	25	30	30	40	40	50
V_{lp}	450	815	600	1120	750	1375	870	1650	1160	2200	1450
C_{lp}	353	453	530	647	653	881	871	1032	1300	1602	1926
G_{lp}	11.5%	5.3%	8.3%	8.7%	8.3%	9.2%	11.3%	12.7%	10.0%	14.6%	11.7%
T_{lp}	0.1s	0.1s	0.1s	0.1s	0.1s	0.1s	0.2s	0.2s	0.2s	0.3s	0.4s
$n + 1$	76	106	101	141	126	176	151	211	201	281	251
C_{sdp}	124	157	164	206	204	256	243	305	323	405	403
C_{ut_1}	408	726	610	1025	833	1364	1075	1695	1634	2486	2334
G_{sdp_1}	8.9%	2.9%	6.9%	6.5%	7.4%	7.7%	10.1%	10.5%	9.2%	14.3%	11.1%
T_{sdp_1}	7s	21s	19s	52s	44s	114s	89s	205s	289s	888s	836s
C_{ut_2}	771	1119	1288	1727	1591	2397	2257	2888	2303	2959	2772
G_{sdp_2}	0.0%	0.0%	0.0%	0.0%	0.0%	0.3%	2.1%	2.9%	2.7%	10.9%	6.5%
Opt_2	10	10	10	10	10	7	3	1	1	0	0
T_{sdp_2}	63s	101s	218s	338s	415s	956s	841s	1741s	1594s	2640s	2654s

Table 3: Results for instances with positive changeover times: general cost structure

Set	B1	B2	B3	B4	B5	B6	B7	B8	B9
P	4	6	4	6	4	6	4	6	4
T	15	15	20	20	25	25	30	30	40
V_{lp}	450	840	600	1120	750	1400	900	1680	1200
C_{lp}	348	439	506	627	692	725	962	1105	1398
G_{lp}	2.4%	0.2%	7.4%	3.0%	16.7%	8.4%	16.7%	11.5%	22.8%
T_{lp}	0.1s	0.1s	0.1s	0.2s	0.1s	0.2s	0.1s	0.2s	0.2s
$n + 1$	91	121	121	161	151	201	181	241	241
C_{sdp}	272	364	361	483	451	577	510	692	678
Cut_1	391	688	573	982	785	1308	1039	1669	1565
G_{sdp1}	11.9%	11.3%	9.7%	9.8%	13.5%	11.8%	9.6%	12.4%	9.3%
T_{sdp1}	13s	37s	27s	86s	58s	168s	147s	333s	437s
Cut_2	1377	1462	1734	2214	2239	2626	2453	2542	2526
G_{sdp2}	0.0%	0.0%	0.6%	0.3%	1.9%	4.9%	2.3%	7.3%	3.9%
Opt_2	10	10	7	8	3	1	2	0	0
T_{sdp2}	205s	393s	640s	962s	921s	2170s	1773s	2231s	2533s

Table 4: Results for instances with positive changeover times: product family cost structure

Set	B1f	B2f	B3f	B4f	B5f	B6f	B7f	B8f	B9f
P	4	6	4	6	4	6	4	6	4
T	15	15	20	20	25	25	30	30	40
V_{lp}	450	815	600	1120	750	1400	900	1680	1200
C_{lp}	346	436	512	630	684	525	975	1112	1400
G_{lp}	3.9%	9.8%	8.0%	7.7%	23.4%	19.4%	23.0%	21.2%	32.6%
T_{lp}	0.1 s	0.1s	0.1s	0.1s	0.1s	0.2s	0.1s	0.2s	0.2s
$n + 1$	91	121	121	161	151	201	181	241	241
C_{sdp}	272	364	361	483	451	577	510	692	678
Cut_1	285	683	576	984	777	1300	1050	1676	1564
G_{sdp1}	34.3%	12.8%	30.6%	39.2%	35.1%	48.7%	21.1%	18.9%	23.7%
T_{sdp1}	12s	40s	35s	87s	61s	176s	165s	390s	469s
Cut_2	1332	1402	2034	2793	2548	2719	2644	2477	2480
G_{sdp2}	0.3%	1.1%	1.1%	6.0%	5.1%	18.6%	4.3%	10.9%	10.9%
Opt_2	9	9	8	2	1	0	0	0	0
T_{sdp2}	238s	452s	600s	2327s	1727s	2750s	2017s	2349s	2464s

Results from Tables 1 and 2 show that, for the instances with zero changeover times, the lower bounds provided by the proposed semidefinite relaxation of the DLSPSD are of significantly improved quality as compared with the ones provided by the strongest linear relaxations known for the problem. Namely, the average gap over the 110 instances A1-A11 involving a general changeover cost structure is decreased from 2.2% with the strengthened linear reformulation to 0.4% with the semidefinite relaxation. The improvement is even more significant for instances A1f-A11f involving a product family cost structure as the average gap over the 110 corresponding instances is decreased from 10.1% to 2.3%.

Similar results can be seen from Tables 3 and 4 for the instances with positive changeover times. Namely, the average gap is decreased from 9.9% to 2.4% for the 90 instances B1-B9 involving a general cost structure and from 16.5% to 6.5% for the 90 instances B1f-B9f involving a family cost structure.

Moreover, we point out that for 166 out of the 200 small-size instances involving up to 4 products and 25 periods, we have $Z_{sdp1} = Z_{DLSLP}$, i.e. the proposed semidefinite relaxation provides the optimal integer solution value so that the residual gap vanishes. This means that for these instances, the discrete problem is solved to exact optimality without resorting to any kind of tree search process. We are not aware of any other previously published relaxation achieving such an accurate approximation of the MIP solution set for the class of discrete lot-sizing problems addressed here.

5.3. Discussion

The improvement in the quality of the lower bounds obtained by using the proposed semidefinite relaxation of the problem might be explained by two main reasons. First, lifting the problem into a higher dimensional space where every possible quadratic term $x_i x_j$ might be used to express the problem constraints enables us to exploit a variety of formulation strengthening techniques which can otherwise not be directly applied when using the linear reformulation of DLSPSD1 considered in [2, 10]. Moreover, by reformulating the problem as a semidefinite program and requiring that the variable matrix X is positive semidefinite, we implicitly add an infinite number of constraints in the problem. Namely, imposing the semidefiniteness of matrix $X \succeq 0$ would be equivalent to incorporating into the linear reformulation an infinite set of constraints of the form $v^t X v \geq 0, \forall v \in \mathbb{R}^{n+1}$.

However, our results are obtained at the expense of significant computation time. This might be first explained by the limitations of available state-of-

the-art semidefinite programming solvers. Solving a semidefinite program is namely more computationally intensive than solving a linear program. As mentioned by [25], currently available SDP interior-point methods work nicely for instances with a matrix size below 200 and less than 2000 constraints whereas instances with a matrix size over 1000 and more than 10000 constraints are considered as impractical. Study of algorithms and development of software capable of solving large-size semidefinite programs is currently an active area of research so that significant progress may be expected in the near future (see e.g. [33]). A second possible explanation for the observed significant computation times could be that, in the proposed solution procedure, we solve a series of semidefinite programs of increasing size during the cutting-plane generation algorithm without being able to use a warm-start strategy. Namely, using the optimal solution of the previous iteration of the cutting-plane generation to reoptimize the problem slightly changed by the addition of some valid inequalities is difficult to implement with interior-point algorithms such as the one embedded in the solver DSDP. It might thus be worth investigating the use of a semidefinite programming solver based on another type of algorithm (such as the spectral bundle method presented in [13]) as this might enable us to more easily exploit a warm-start strategy during the cutting-plane generation.

6. Conclusion

We studied the discrete lot-sizing and scheduling problem with sequence-dependent changeover costs and times. This optimization problem can be formulated as a quadratically constrained quadratic binary program. We proposed in the present paper to compute a lower bound of the optimal integer solution value of the problem by carrying out a semidefinite relaxation. To achieve this, we not only exploited reformulation and strengthening techniques recently proposed in the SDP literature for generic (0-1) quadratic binary problems but also used problem-specific information such as the polyhedral representation of single-product discrete lot-sizing problems. The results of our computational experiments show that, in terms of solution quality, the proposed approach compares well with the best MILP strengthening techniques known for the problem. It provides lower bounds of significantly improved quality, especially for the instances featuring a product family cost structure, and is capable of completely closing the gap between the lower bound and the optimal integer solution value for a significant proportion of

the small-size instances. However, due to the limitations of available state-of-the-art semidefinite programming solvers, these results are obtained at the expense of significant computation times so that it does not seem possible for the time being to solve large-size industrial instances with the proposed approach.

The present work can thus be viewed as a first step towards applying SDP models and tools to discrete lot-sizing problems. Sure, the reported computation times are still significant, however future improvements in the computational efficiency of the SDP technology are likely to occur. In such a perspective, the methodology proposed here might serve as a basis for competitive SDP-based approaches to lot-sizing.

Acknowledgement

The work described in this paper was funded by the French National Research Agency through its research funding program JCJC2011 for young researchers (project ANR-11-JS0002-01 LotRelax). The authors would also like to thank two anonymous referees for their detailed reviews that helped improving an initial version of this paper.

- [1] M.F. Anjos, A. Kennings and A. Vannelli. A semidefinite optimization approach of the single-row layout problem with unequal dimensions. *Discrete Optimization*, vol. 2, pp 113-122, 2005.
- [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] S.J. Benson, Y. Ye and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, vol. 10(2), pp 443-461, 2000.
- [4] 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.
- [5] 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.

- [6] A. Engau. Recent progress in interior-point methods: cutting-plane methods and warm starts. *Handbook of semidefinite, conic and polynomial optimization*, International Series in Operations Research and Management Science, Springer, pp 471-488, 2012.
- [7] G.D. Eppen and R.K Martin. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, vol. 35(6), pp 832-848, 1987.
- [8] 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.
- [9] B. Fleischmann. The discrete lot sizing and scheduling problem. *European Journal of Operational Research*, vol. 44, pp 337-348,1990.
- [10] 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.
- [11] 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.
- [12] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, vol.42(6), pp 1115-1145, 1995.
- [13] C. Helmberg, F. Rendl and R. Weismantel. A semidefinite programming approach to the quadratic knapsack problem. *Journal of Combinatorial Optimization*, vol. 4(2), pp 197-215, 2000.
- [14] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal of Optimization*, vol. 10(3), 673-696, 2000.
- [15] C. Helmberg, 2000. Semidefinite programming for combinatorial optimization. Habilitationsschrift, TU Berlin.

- [16] C. Helmberg. Semidefinite programming. *European Journal of Operational Research*, vol. 137(3), pp 461-482, 2002.
- [17] 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.
- [18] R. Jans and Z. Degraeve. Modelling industrial lot sizing problems: a review. *Industrial Journal of Production Research*, ,vol. 46(6), pp 1619-1643, 2008.
- [19] C. Lemarechal and F. Oustry. Semidefinite relaxations and lagrangian duality with application to combinatorial optimization. RR-3710, INRIA Rhones-Alpes, ZIRST-655 avenue de l'Europe, F-38330 Montbonnot Saint-Martin, June 1999.
- [20] B. Lovàsz and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, vol. 1, pp 166-190, 1991.
- [21] S. Mhanna and R. Jabr. Application of semidefinite programming relaxation and selective pruning to the unit commitment problem, *Electric Power System Research*, vol. 90, pp 85,92, 2012.
- [22] Y. Pochet and L.A. Wolsey. Production planning by mixed integer programming, Springer Science, 2006.
- [23] S. Poljak, F. Rendl and H. Wolkowicz. A recipe for semidefinite relaxation of (0-1) quadratic programming. *Journal of Global Optimization*, vol. 7, pp 51-73, 1995.
- [24] J. Povh and F. Rendl. Copositive and semidefinite relaxations of the quadratic assignment problem. *Discrete Optimisation*, vol. 6, pp 231-241, 2009.
- [25] F. Rendl. Semidefinite relaxations for partitioning, assignment and ordering problems. *4OR-Quarterly Journal of Operations Research*, vol. 10, pp 321-346, 2012.
- [26] F. Roupin. From linear to semidefinite programming: an algorithm to obtain semidefinite relaxations for bivalent quadratic problems. *Journal of Combinatorial Optimization*, vol. 8, pp 469-493, 2004.

- [27] 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.
- [28] H.D. Sherali and W.P. Adams. A hierarchy of relaxations between continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, vol. 3 (3), pp 411-430, 1990.
- [29] 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.
- [30] M. Skutella. Convex quadratic and semidefinite programming relaxations in scheduling. *Journal of the ACM*, vol 48 (2), pp 206-242, 2001.
- [31] 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.
- [32] H.M. Wagner and T.M. Whitin. Dynamic version of the economic lot size model. *Management Science*, vol. 5(1), pp 89-96, 1958.
- [33] M. Yamashita, K.Fujisawa, M. Fukuda, K. Kobayashi, K. Nakata and M. Nakata. Latest Developments in the SDPA Family for Solving Large-Scale SDPs. *Handbook on Semidefinite, Conic and Polynomial Optimization*. International series in Operations Research and Management Science, Springer, pp 687-714, 2012.
- [34] Q. Zhao, S.E. Kharisch, F. Rendl and H. Wolkovicz. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, vol. 2, pp 71-109.