SYMMETRY-EXPLOITING CUTS FOR A CLASS OF MIXED-0/1 SECOND-ORDER CONE PROGRAMS

SARAH DREWES AND SEBASTIAN POKUTTA

Abstract. We will analyze mixed-0/1 second-order cone programs where the continuous and binary variables are solely coupled via the conic constraints. We devise a cutting-plane framework based on an implicit Sherali-Adams reformulation. The resulting cuts are very effective as symmetric solutions are automatically cut off and each equivalence class of 0/1 solutions is visited at most once. Further, we present computational results showing the effectiveness of our method and briefly sketch an application in optimal pooling of securities.

1. Introduction

Mixed-integer second-order cone programming, as one of the rather tractable problem classes of mixed-integer nonlinear programming, gained strong interest in recent years. Due to their structure and convexity, the underlying relaxations can be solved in polynomial time to ϵ-optimality. Mixed-integer second-order cone programs are usually solved via outer approximation algorithms or within a branch-and-cut framework (see e.g., [22], [11] or [6] for an overview).

Here we consider a restricted class of mixed-integer second-order cone programs where the integral variables are binary and the only coupling of binary and continuous variables occurs in the conic constraints. For this problem class, we derive cutting-planes based on an implicit Sherali-Adams reformulation of the problem and subsequent application of subgradient based cuts (see e.g., [6, 13, 9]).

The Sherali-Adams hierarchy (cf. [19]) is a well-known tool to generate successively tighter linear approximations of the integral hull of a polytope. Recently, it regained strong interest, in particular in connection with inapproximability of certain combinatorial problems as well as in the context of algebraic geometry (see e.g., [15, 16, 8, 17]). We will use the Sherali-Adams hierarchy on an implicit level here by interpreting the second-order cone program as the intermediate step of the Sherali-Adams procedure. In contrast to the Sherali-Adams hierarchy, which is used for convexification, outer approximation algorithms (see [4, 13, 12, 20]) are used to decompose a problem into a master problem and subproblems. Given an optimal solution to the master problem, one solves the subproblem and derives cutting planes which in turn are added back to the master problem and the process repeats.

The first part of our reformulation exploits the fact that for a binary variable x it holds $x^2 - x = 0$ in order to strengthen the formulation. Applying this reformulation technique, we disentangle the binary variables from the continuous variables in the conic constraints. As a result we obtain a reformulation where the binary variables are aggregated in the linear parts of the conic conditions. In a second step we derive a family of cutting planes based on subgradients of the constraint functions which will be used in an outer approximation framework. While generalized Benders cut and classical outer approximation are often outperformed by more advanced methods, the proposed combination works very well in this case. In fact, their effectiveness stems from their ability to cut off equivalence classes of symmetric

Date: April 16, 2014/Draft.

2000 *Mathematics Subject Classification.* Primary 90C11; Secondary 90C57 90C25.

Key words and phrases. Mixed-Integer Nonlinear Programming, Cutting Planes, Second Order Cone Programming, Outer Approximation.

An extended abstract of this paper appeared in the proceedings of the International Symposium on Combinatorial Optimization 2010.

solutions, thus vastly reducing the number of outer approximation iterations. We apply these cuts and the resulting decomposition to solve a pooling problem arising in portfolio optimization in finance.

Related work. Cutting-planes for mixed-integer second-order cone programs have been extensively investigated. For example in [7] lift-and-project based cuts for mixed 0/1 conic programming problems have been studied. In [2], Gomory mixed-integer rounding cuts for second-order cone programs have been devised and in [3] lifting for conic mixed-integer programming was investigated. A branch-and-cut based method for convex mixed 0/1 programming was outlined in [21] and in [10] lift-and-project based cutting-planes as well as subgradient based outer approximations have been applied to solve mixed-integer second-order cone programs. A lifted linear programming branch-and-bound algorithm for second-order cone programs, where second-order cone constraints are approximated via linear ones, was outlined in [22]. Our approach is different as we consider a special class of mixed 0/1 second-order cone programs whose structure we exploit.

Our contribution. We propose a reformulation technique and resulting cutting planes for a special class of second-order cone programs where the coupling of continuous and binary variables occurs only in the conic constraints. In a first step the formulation is strengthened by applying the Sherali-Adams closure in an implicit fashion. For this strengthened reformulation we derive strong cutting planes, which are applied in an outer approximation framework and evaluated in computational experiments.

Outline. In Section 2 we briefly recall the necessary notation, followed by the reformulation of the considered problem class in Section 3. We then introduce the cutting-plane framework and derive cutting-planes in Section 4 based on this reformulation. We conclude with a pooling problem arising in portfolio optimization in Section 5, computational results in Section 6, and some final remarks in Section 7.

2. Preliminaries

For $n \in \mathbb{N}$ we write $[n] := \{1, \dots, n\}$ and for a set $J \subseteq [n]$, we define $\bar{J} := [n] \setminus J$ to be the *complement* of J in $[n]$. All other notation is standard as to be found in [1, 10].

We will consider the following class of second-order cone programs:

Definition 2.1. A *weakly-coupled 0/1 second-order cone program* (wSOC) has the form:

$$\begin{aligned}
&\text{(wSOC)} && \min c_J x_J + c_{\bar{J}} x_{\bar{J}} \\
&(2.1) && \text{s.t. } A_J x_J \le b_J \\
&(2.2) && A_{\bar{J}} x_{\bar{J}} \le b_{\bar{J}} \\
&(2.3) && \left\| (W_\ell x_{J_\ell}, Q_\ell x_{\bar{J}_\ell}) \right\|_2 \le \alpha_\ell x_\ell^0 + \tau_\ell \quad \forall\, \ell \in [k] \\
&(2.4) && x_{\bar{J}} \in \mathscr{K} \\
&(2.5) && x_{\bar{J}} \in \mathbb{R}^{|\bar{J}|} \\
&(2.6) && x_J \in \{0,1\}^{|J|} \\
&(2.7) && x_\ell^0 \in \{0,1\} \quad \forall\, \ell \in [k],
\end{aligned}$$

with $n \in \mathbb{N}$, $J \subseteq [n]$, $k \in \mathbb{N}$, $A_J \in \mathbb{R}^{m_J \times |J|}$, $W_\ell \in \mathbb{R}^{m_{J_\ell} \times |J_\ell|}$, $A_{\bar{J}} \in \mathbb{R}^{m_{\bar{J}} \times |\bar{J}|}$, $Q_\ell \in \mathbb{R}^{m_{\bar{J}_\ell} \times |\bar{J}_\ell|}$, $b_J \in \mathbb{R}^{m_J}$, $b_{\bar{J}} \in \mathbb{R}^{m_{\bar{J}}}$, $c = (c_J, c_{\bar{J}}) \in \mathbb{R}^n$, and $\alpha_\ell, \tau_\ell \in \mathbb{R}$ for all $\ell \in [k]$. Further let $J_\ell \subseteq J$ and $\bar{J}_\ell \subseteq \bar{J}$ for all $\ell \in [k]$. The index set J_ℓ denotes the binary variables of the ℓ-th second-order cone constraint and similarly $\bar{J}_\ell$ denotes the continuous variables. Moreover, $\mathscr{K} \subseteq \mathbb{R}^{|\bar{J}|}$ in Condition (2.4) denotes a convex feasible region arising from additional second-order cone constraints on the continuous variables $x_{\bar{J}}$.

Note that additional second-order cone constraints solely in the binary variables can be rewritten as linear ones with the same feasible solutions using standard techniques and thus can be considered to be included in (2.1); appearing mixed terms can be linearized, e.g., with McCormick linearizations. The second-order cone conditions (2.3) resemble a typical on/off constraint where the leading cone variable is given by an affine-linear function in a 0/1 variable. We slightly abuse notation by using $\bar{J}_\ell$ to refer to the index subset of continuous variables contained in the ℓ-th second-order cone constraint, i.e., we have $J_\ell \subseteq J$, $\bar{J}_\ell \subseteq \bar{J}$, and $J_\ell \mathbin{\dot\cup} \bar{J}_\ell \subseteq [n]$ but equality does not necessarily have to hold. The only coupling of continuous and binary variables occurs in the conic constraints (2.3).

We will now reformulate the program so that this coupling is shifted from the conic constraints to some additional linear ones. This will allow us later to derive strong cutting-planes for an outer approximation algorithm.

3. Sherali-Adams based reformulation

In this section we will show how the conic constraints (2.3) can be reformulated so that no integral variables are contained in the nonlinear part, i.e., within the norm part of the constraint. This reformulation mimics the Sherali-Adams procedure (see [19]) but is implicit in the sense, that the program can be understood as one to which the lifting of the Sherali-Adams procedure has been partially applied. We gain a strengthened reformulation of the conic constraints by performing the remaining relinearization step on a subset of the variables.

Throughout the section, for brevity, we shall consider a single conic constraint of the form

$$\left\|(W_\ell x_{J_\ell}, Q_\ell x_{\bar{J}_\ell})\right\|_2 \le \alpha_\ell x^0_\ell + \tau_\ell \tag{3.1}$$

as given in (2.3) for some $\ell \in [k]$. For the remainder of this section we drop the index ℓ as we consider a single constraint only and assume $W \in \mathbb{R}^{m_J \times |J|}$.

We add new variables z_{lj} with $l < j \in J$ that will represent the multiplication of x_l with x_j whenever, $w_{il}w_{ij} \neq 0$ for some $i \in [m_J]$. This lifting step corresponds to the lifting performed in the Sherali-Adams procedure.

Lemma 3.1. *Let*

$$\left\|(W x_J, Q x_{\bar{J}})\right\|_2 \le \alpha x^0 + \tau, \quad x_J \in \{0,1\}^{|J|},\, x^0 \in \{0,1\}$$

be as above. Then this constraint is equivalent to

$$\left\|Q x_{\bar{J}}\right\|_2^2 \le \tau^2 + (\alpha^2 + 2\alpha\tau)x^0 - \sum_{i\in[m_J]} \left(\sum_{l\in J} w_{il}^2 x_l + \sum_{l<j\in J} 2 w_{il} w_{ij} z_{lj} \right) \tag{3.2}$$

$$0 \le a x^0 + \tau \tag{3.3}$$

$$z_{lj} \le x_j \quad \forall\ j < l \in J \text{ with } w_{il}w_{ij} \neq 0 \text{ for some } i \in [m_J] \tag{3.4}$$

$$z_{lj} \le x_l \quad \forall\ j < l \in J \text{ with } w_{il}w_{ij} \neq 0 \text{ for some } i \in [m_J] \tag{3.5}$$

$$z_{lj} \ge x_j + x_l - 1 \quad \forall\ j < l \in J \text{ with } w_{il}w_{ij} \neq 0 \text{ for some } i \in [m_J] \tag{3.6}$$

$$x_i \in \{0,1\} \quad \forall\ i \in J \tag{3.7}$$

$$z_{lj} \in \mathbb{R}_+ \quad \forall\ j < l \in J \tag{3.8}$$

Proof. Observe that $\left\|(Q x_{\bar{J}}, W x_J)\right\|_2 \le \alpha x^0 + \tau$ is equivalent to $\left\|Q x_{\bar{J}}\right\|_2^2 + \left\|W x_J\right\|_2^2 \le (\alpha x^0 + \tau)^2$ with $\alpha x^0 + \tau \ge 0$. We will now rewrite $\left\|W x_J\right\|_2^2$ to obtain the desired form. With $W := (w_1, \ldots, w_{m_J})^T =$

$(w_{ij})_{i\in[m_J],j\in J}$ we have

$$\left\|W x_J\right\|_2^2 = \sum_{i\in[m_J]} (w_i x)^2 = \sum_{i\in[m_J]} \left(\sum_{j\in J} w_{ij} x_j\right)^2 .$$

Applying the binomial formula to $\left(\sum_{j\in J} w_{ij} x_j\right)^2$, we obtain

$$\left(\sum_{j\in J} w_{ij} x_j\right)^2 = \sum_{l\in J} w_{il}^2 x_l^2 + \sum_{l<j\in J} 2 w_{il} w_{ij} x_l x_j .$$

As x_J corresponds to binary solutions, it holds that $x_i^2 - x_i = 0$ and one can further relinearize $x_l x_j =: z_{lj}$ so the above can be rewritten as

$$\left(\sum_{j\in J} w_{ij} x_j\right)^2 = \sum_{l\in J} w_{il}^2 x_l + \sum_{l<j\in J} 2 w_{il} w_{ij} z_{lj} .$$

Similarly we rewrite $(\alpha x^0 + \tau)^2$ as $\tau^2 + (\alpha^2 + 2\alpha\tau)x_0$ using $x_0^2 = x_0$. We finally add the following constraints to ensure that $z_{lj} = x_l x_j$ holds for all valid solutions:

$$\begin{aligned}
&z_{lj} \le x_j \quad \forall\, j < l \in J \text{ with } w_{il} w_{ij} \neq 0 \text{ for some } i \in [m_J],\\
&z_{lj} \le x_l \quad \forall\, j < l \in J \text{ with } w_{il} w_{ij} \neq 0 \text{ for some } i \in [m_J],\\
&z_{lj} \ge x_j + x_l - 1 \quad \forall\, j < l \in J \text{ with } w_{il} w_{ij} \neq 0 \text{ for some } i \in [m_J].
\end{aligned}$$

This finishes the proof. □

Observe that only for those binary variables that occur in the conic constraints we have to introduce additional variables. Thus, whenever only a few binary variables are present, we have to add a rather modest number of extra variables. Furthermore, note that the relations ensuring $x_i x_j = z_{ij}$ have to be included only once, even if the same variable combination occurs in different conic constraints. The resulting reformulation can be expressed as a second-order cone program by a standard transformation (see e.g., [1]). For the reformulation at hand it is crucial that only binary variables occur in the right hand side of (3.1), since their squares can be substituted by linear terms (i.e., $x = x^2$). Otherwise a reformulation similar to Lemma 3.1 is not necessarily convex. Whenever $\alpha = 0$, the relaxation of the conic constraint in Lemma 3.1 obtained for $x \in [0,1]^n$ is tighter than the original formulation due to $x^2 \le x$ for all $x \in [0,1]$; this can be easily seen by comparing both formulations. For the ease of presentation we will assume $\alpha_\ell = 0$, $\tau_\ell \in \mathbb{R}_+$ for all $\ell \in [k]$, and we drop the constraints (2.4) for the remainder of this article. The general case follows readily by substitution.

We now substitute the coupling conic constraints (2.3) in (wSOC) by the above reformulation. This reformulation will be called the *strong quadratic equivalent (SQE)*:

$$\text{(SQE)} \qquad \min\ c_J x_J + c_{\bar{J}} x_{\bar{J}}$$

$$\text{(3.9)} \qquad \text{s.t. } A_J x_J \leq b_J$$

$$\text{(3.10)} \qquad A_{\bar{J}} x_{\bar{J}} \leq b_{\bar{J}}$$

$$\text{(3.11)} \qquad \left\| Q_\ell x_{\bar{J}_\ell} \right\|_2^2 \leq \tau_\ell^2 - \sum_{i \in [m_{J_\ell}]} \left(\sum_{l \in J_\ell} \left(w_{il}^\ell \right)^2 x_l + \sum_{l,j \in J_\ell, l<j} 2 w_{il}^\ell w_{ij}^\ell z_{lj} \right) \quad \forall\ \ell \in [k]$$

$$\text{(3.12)} \qquad z_{lj} \leq x_j \quad \forall\ j < l \in J : w_{il}^\ell w_{ij}^\ell \neq 0 \text{ for some } i \in [m_{J_\ell}], \ell \in [k]$$

$$\text{(3.13)} \qquad z_{lj} \leq x_l \quad \forall\ j < l \in J : w_{il}^\ell w_{ij}^\ell \neq 0 \text{ for some } i \in [m_{J_\ell}], \ell \in [k]$$

$$\text{(3.14)} \qquad z_{lj} + 1 \geq x_j + x_l \quad \forall\ j < l \in J : w_{il}^\ell w_{ij}^\ell \neq 0 \text{ for some } i \in [m_{J_\ell}], \ell \in [k]$$

$$\text{(3.15)} \qquad z_{lj} \in \mathbb{R}_+ \quad \forall\ j < l \in J$$

$$\text{(3.16)} \qquad x_J \in \{0,1\}^J$$

$$\text{(3.17)} \qquad x_{\bar{J}} \in \mathbb{R}^{\bar{J}}$$

$$\text{(3.18)} \qquad h \in \mathbb{R}_+^k$$

Note that indeed (wSOC) and (SQE) are equivalent by projecting (SQE) onto the x variables. In fact (SQE) is an extended formulation of (wSOC). Moreover, the z variables are completely determined by x_J and thus we will later consider subproblems where we fix x_J only; the fixing of the z variables is implied then.

4. Cutting-planes for Weakly-Coupled 0/1 SOCPs (wSOC)

We will now derive strong cutting-planes for weakly-coupled mixed 0/1 second-order cone programs that will be used later in an outer approximation framework (see Section 4.5). Our family of cuts is based on outer approximation cuts and in their final form they will resemble the generalized Benders cuts (see also e.g., [13], [6]) that we briefly recall in Section 4.1 and will be tailored to the strong quadratic equivalent in Section 4.2.

Consider the quadratic part (3.11) of (SQE) and define auxiliary variables h_ℓ via

$$\text{(4.1)} \qquad h_\ell := \tau_\ell^2 - \sum_{i \in [m_{J_\ell}]} \left(\sum_{l \in J_\ell} \left(w_{il}^\ell \right)^2 x_l + \sum_{l,j \in J_\ell, l<j} 2 w_{il}^\ell w_{ij}^\ell z_{lj} \right)$$

where $J_\ell \subseteq J$ as in Definition 2.1. Observe that h_ℓ can attain only values between the bounds determined by the right hand side of (4.1). The particular assignment of the binary variables x_j, z_{lj} with $j, l \in J_\ell$ and $l < j$ does not change the resulting conic constraint (or more precisely its strong quadratic equivalent) as long as the associated h_ℓ remains unchanged, as we require

$$\text{(4.2)} \qquad h_\ell \geq \left\| Q_\ell x_{\bar{J}_\ell} \right\|_2^2 .$$

Hence, to determine a feasible solution to (3.11), it suffices to investigate the solutions of the continuous conic constraints (4.2) for potential assignments of the variables h_ℓ. The cut derived in the following exploits this fact, i.e., that the conic constraint does only depend on the h_ℓ variables and not the particular assignment to the binary variables x_J.

For the sake of completeness we now briefly recall the concept of an outer approximation algorithm as proposed for example by [5], [11], [12], [18] or [14]. While executing such an algorithm iteratively, integer feasible solutions of a linear outer approximation problem are computed and nonlinear subproblems are solved that arise from the given mixed-integer nonlinear problem (here: (wSOC)) by fixing the integral part of the variables. The solutions of these nonlinear subproblems are feasible solutions of (wSOC) and thus provide upper bounds. Moreover, they give rise to successively tighter linear outer approximations by inducing certain outer approximation cuts. These linear outer approximations are usually based on (sub)gradient cuts arising from the Karush-Kuhn-Tucker (KKT) optimality conditions of an optimal solution to the nonlinear subproblems. In the case that fixing the integer part of (wSOC) leads to an infeasible subproblem, a feasibility problem is solved which provides similar linear approximations.

4.1. **Outer approximation cuts.** We briefly summarize the gradient based linearizations that can be used in an outer approximation framework for mixed integer nonlinear optimization problems. For a more extensive introduction the reader is referred to [6] and references contained therein. Consider a general mixed integer nonlinear optimization problem

$$\text{(MiNLP)} \qquad \begin{aligned} &\min \eta \\ &\text{s.t. } f(x) \le \eta \\ &\quad g_i(x) \le 0, \ \forall i \in I \\ &\quad Ax \le b \\ &\quad x_j \in \mathbb{Z}, \ j \in J, \end{aligned}$$

where f and g_i with $i \in I$ are convex and sufficiently smooth functions, $A \in \mathbb{R}^{m,n}$ and $b \in \mathbb{R}^m$. Observe that the objective function is moved into the constraints and upper bounded by η; minimizing η minimizes the objective function. This will be helpful for deriving the cutting-planes. For a *feasible* assignment $\bar{x}_J$ for the integer variables in (MiNLP) let

$$(\text{NLP}(\bar{x}_J))$$

denote the program where we fix the integer variables $x_J = \bar{x}_J$. Assume (NLP($\bar{x}_J$)) satisfies a constraint qualification and let $\bar{x}_{\bar{J}}$ be an optimal solution of (NLP($\bar{x}_J$)). Then due to convexity, the inequalities

$$g_i(\bar{x}) + \nabla g_i(\bar{x})^T(x - \bar{x}) \le 0 \tag{4.3}$$

and

$$f(\bar{x}) + \nabla f(\bar{x})^T(x - \bar{x}) \le \eta \tag{4.4}$$

are valid linearizations of the nonlinear constraints and so they are satisfied by any point x in the feasible region of (MiNLP). Using the KKT conditions of (NLP($\bar{x}_J$)), it can be shown that every x with $x_J = \bar{x}_J$ that satisfies (4.3) for all i and (4.4) induces an objective function value $f(x)$ that cannot improve on the objective function value $f(\bar{x})$. In other words, the constraints (4.3) and (4.4) capture in the outer approximation the full consequence of fixing the binary variables in (MiNLP).

In case the subproblem (NLP($\bar{x}_J$)) does not have a feasible solution $x_{\bar{J}}$, the linearizations arise from the solution of a feasibility version of (NLP($\bar{x}_J$)) where we relax the nonlinear constraints $g_i(x) \le 0$ to $g_i(x) \le u$ for $i \in I$ by $u \ge 0$ and minimize the violation u. With $(\bar{u}, \bar{x})$ solving that feasibility problem we add the linearizations (4.3) and (4.4) in that solution $\bar{x}$. Similar to the feasible case, the KKT conditions of the feasibility problem can be used to show that every x with $x_J = \bar{x}_J$ violates (4.3) and (4.4). These two properties ensure that the outer approximation algorithm converges and every (potential) integral assignment is visited at most once. For details of the general case we refer to [12] and [11].

4.2. **Binary symmetric cut.** We will now derive gradient based outer approximation cuts based on (4.3) and (4.4) that exploit the fact that the conic constraints (or their strong quadratic equivalent) depend only on the h variables as discussed before. Therefore, the cut constrains all integral assignments that induce the same h values simultaneously and thus accounts for this symmetry. We distinguish feasible and infeasible subproblems.

4.2.1. *Optimality cut.* Given an integral assignment $\bar{x}_J$, we consider the nonlinear subproblem

$$(\mathrm{NLS}(\bar{x}_J)) \qquad \text{(SQE) together with } x_J = \bar{x}_J$$

which is obtained from (SQE) by fixing the binary variables to $\bar{x}_J$. We put $f(x) = c^T x$ and for the nonlinear constraints g_i we can assume without loss of generality that $Q_\ell = I$ is the identity matrix of appropriate dimension. The nonlinear constraints in (SQE) are then given by

$$g_\ell(x) = -\tau_\ell^2 + \sum_{i \in [m_{J_\ell}]} \left(\sum_{l \in J_\ell} \left(w_{il}^\ell\right)^2 x_l + \sum_{l < j \in J_\ell} 2 w_{il}^\ell w_{ij}^\ell z_{lj} \right) + \left\| x_{\bar{J}_\ell} \right\|_2^2 \leq 0 \quad \forall\, \ell \in [k]. \tag{4.5}$$

with gradients

$$(\nabla_\zeta g_\ell(x)) = \begin{cases} \sum_{i \in [m_{J_\ell}]} \left(w_{il}^\ell\right)^2, \text{ if } \zeta = x_l \text{ and } l \in J_\ell, \\ \sum_{i \in [m_{J_\ell}]} 2 w_{il}^\ell w_{ij}^\ell, \text{ if } \zeta = z_{lj} \text{ and } l < j \in J_\ell, \\ 2 x_l, \text{ if } \zeta = x_l \text{ and } l \in \bar{J}_\ell. \end{cases} \quad \text{for } \ell \in [k]. \tag{4.6}$$

In the context of outer approximation, it suffices to consider only active constraints, hence we can assume $g_\ell(\bar{x}) = 0$. The linearizations corresponding to (4.3) are then $\nabla g_\ell(\bar{x})(x - \bar{x}) \leq 0$, namely

$$\sum_{l \in J_\ell} \left(\sum_{i \in [m_{J_\ell}]} \left(w_{il}^\ell\right)^2 (x_l - \bar{x}_l) + \sum_{i \in [m_{J_\ell}]} 2 w_{il}^\ell w_{ij}^\ell (z_\ell - \bar{z}_\ell) \right) + \sum_{l \in \bar{J}_\ell} 2(x_l - \bar{x}_l) \leq 0 \text{ for } \ell \in [k]. \tag{4.7}$$

In the following we show that assignments to the integral variables that induce the same h values (cf. condition (4.1)) are cut off by the above inequality.

Lemma 4.1. *Let $\bar{x}_J$ be an integer assignment with induced $\bar{h} = (\bar{h}_\ell)_\ell$ for which (*NLS$(\bar{x}_J)$*). Let $(\bar{x}, \bar{z})$ be an optimal solution to (*NLS$(\bar{x}_J)$*). Then every $x \in \mathbb{R}^n$ satisfying* (4.7) *with an integer assignment x_J inducing $h = (h_\ell)_\ell$ with $h = \bar{h}$ as defined in* (4.1) *satisfies*

$$c_{\bar{J}}^T x_{\bar{J}} \geq c_{\bar{J}}^T \bar{x}_{\bar{J}}. \tag{4.8}$$

Proof. We prove the claim by deriving the Generalized Benders cut which is induced by the inequalities (4.7). For this, we first multiply each inequality (4.7) by its Lagrange multiplier corresponding to the KKT system of (NLS$(\bar{x}_J)$), sum up the resulting inequalities and split up the gradients into fractional and integer parts yielding

$$\sum_{\ell \in [k]} \bar{\mu}_\ell \nabla_{\bar{J}} g_\ell(\bar{x})^T (x_{\bar{J}} - \bar{x}_{\bar{J}}) + \sum_{\ell \in [k]} \bar{\mu}_\ell \nabla_J g_\ell(\bar{x})^T (x_J - \bar{x}_J) \leq 0. \tag{4.9}$$

Substituting the KKT conditions of (NLS$(\bar{x}_J)$), namely $-c_{\bar{J}} = \sum_{\ell \in [k]} \bar{\mu}_\ell \nabla_{\bar{J}} g_\ell(\bar{x})$, implies

$$\sum_{\ell \in [k]} \bar{\mu}_\ell \nabla_J g_\ell(\bar{x})^T (x_J - \bar{x}_J) \leq c_{\bar{J}}^T (x_{\bar{J}} - \bar{x}_{\bar{J}}) \tag{4.10}$$

which is equivalent to

$$\sum_{\ell \in [k]} \bar{\mu}_\ell \sum_{i \in [m_{J_\ell}]} \left(\sum_{l \in J_\ell} \left(w_{il}^\ell\right)^2 (x_l - \bar{x}_l) + \sum_{i, l < j \in J_\ell} 2 w_{il}^\ell w_{ij}^\ell (z_{lj} - \bar{z}_{lj}) \right) \leq c_{\bar{J}}^T (x_{\bar{J}} - \bar{x}_{\bar{J}}). \tag{4.11}$$

With (4.1) we have

$$h_\ell = \tau_\ell^2 - \sum_{i \in [m_{J_\ell}]} \left(\sum_{l \in J_\ell} \left(w_{il}^\ell\right)^2 x_l + \sum_{l < j \in J_\ell} 2 w_{il}^\ell w_{ij}^\ell z_{lj} \right) \text{ for all } \ell \in [k]$$

and

$$\bar{h}_\ell = \tau_\ell^2 - \sum_{i\in[m_{J_\ell}]} \left(\sum_{l\in J_\ell} \left(w_{il}^\ell\right)^2 \bar{x}_l + \sum_{l,j\in J_\ell, l<j} 2w_{il}^\ell w_{ij}^\ell \bar{z}_{lj} \right) \text{ for all } \ell \in [k],$$

and therefore we obtain that (4.11) is equivalent to

$$\sum_{\ell\in[k]} \bar{\mu}_\ell \left(h_\ell - \bar{h}_\ell\right) \geq c_{\bar{J}}^T (\bar{x}_{\bar{J}} - x_{\bar{J}}). \tag{4.12}$$

The result now readily follows by substitution. □

Lemma 4.1 states that whenever we add (4.7) for a given optimal solution $(\bar{x}, \bar{z})$ of (NLS($\bar{x}_J$)), to the outer approximation problem all subsequent solutions (x, z) to the outer approximation problem that satisfy $\bar{h} = h$ also satisfy $c_{\bar{J}}^T x_{\bar{J}} \geq c_{\bar{J}}^T \bar{x}_{\bar{J}}$. So either the current solution is optimal or we obtain a new solution with h values different from $\bar{h}$. It follows that any h configuration that leads to a feasible subproblem will be visited at most once. The same holds true for those h configurations that lead to infeasible subproblems as we will see soon. This property is essential for the performance of the algorithm: a significantly smaller amount of iterations have to be performed. In fact rather than visiting every single feasible integral assignment, we only have to consider the equivalence classes induced by h.

It follows directly from the proof that it suffices to add the aggregated cut (4.12) instead of the linearizations (4.7) to achieve (4.8) for all solutions corresponding to $\bar{h}$.

Corollary 4.2. *Let $\bar{x}_J$ be an integer assignment, for which (*NLS($\bar{x}_J$)*) is feasible. Let $(\bar{x}, \bar{z})$ be an optimal solution to (*NLS($\bar{x}_J$)*). Then every $x \in \mathbb{R}^n$ satisfying*

$$\sum_{\ell\in[k]} \bar{\mu}_\ell \left(h_\ell - \bar{h}_\ell\right) \geq c_{\bar{J}}^T (\bar{x}_{\bar{J}} - x_{\bar{J}}), \tag{BSC-O}$$

with an integer assignment x_J inducing h with $h = \bar{h}$ as defined in (4.1) *satisfies $c_{\bar{J}}^T x_{\bar{J}} \geq c_{\bar{J}}^T \bar{x}_{\bar{J}}$. Here, $\bar{\mu}_\ell$ are the Lagrange multipliers associated with the constraints (3.11) satisfying the KKT system of the nonlinear subproblem (*NLS($\bar{x}_J$)*).*

We refer to the Generalized Benders cut (BSC-O) as the *optimality version of the binary symmetric cut* in which the weighted sum of changes in the leading conic variables h_ℓ with $\ell \in [k]$ restricts the change in the continuous part of the objective function. Observe that the cut is only expressed in the h variables and thus independent of the specific configuration of binary variables.

4.2.2. *Feasibility cut.* Let $\bar{x}_J$ be an integer assignment so that (NLS($\bar{x}_J$)) is infeasible. We then solve the feasibility problem of (SQE), which is given by

(NLSF($\bar{x}_J$)) $\quad \min u$

(4.13) $\quad s.t.$ (SQE) where (3.11) is relaxed to

$$\left\| Q_\ell x_{\bar{J}_\ell} \right\|_2^2 \leq u + \tau_\ell^2 - \sum_{i\in[m_{J_\ell}]} \left(\sum_{l\in J_\ell} \left(w_{il}^\ell\right)^2 x_l + \sum_{l,j\in J_\ell, l<j} 2w_{il}^\ell w_{ij}^\ell z_{lj} \right) \quad \forall\, \ell \in [k], \tag{4.14}$$

$$u \geq 0 \tag{4.15}$$

$$x_J = \bar{x}_J. \tag{4.16}$$

Note that $u > 0$ holds for any feasible solution of (NLSF($\bar{x}_J$)) if (NLS($\bar{x}_J$)) is infeasible. The above relaxation is similar to changing the auxiliary variables (4.1) to

$$h_\ell = u + \tau_\ell^2 - \sum_{i \in [m_{J_\ell}]} \left(\sum_{l \in J_\ell} \left(w_{il}^\ell \right)^2 x_l + \sum_{l,j \in J_\ell, l<j} 2 w_{il}^\ell w_{ij}^\ell z_{lj} \right) \tag{4.17}$$

for all $\ell \in [k]$ as these imply a relaxation of (3.11). We add linearizations (4.3) of the constraints (3.11) to the outer approximation problem that correspond to the optimal solution of (NLSF($\bar{x}_J$)), hence we add

$$\bar{u} + \nabla g_\ell(\bar{x})^T (x - \bar{x}) \leq 0 \quad \forall \ell \in [k]. \tag{4.18}$$

Note that the active constraints of (NLSF($\bar{x}_J$)) satisfy $g_\ell(\bar{x}) = \bar{u}$.

Lemma 4.3. *Let $\bar{x}_J$ be an integer assignment for which (*NLS($\bar{x}_J$)*) is infeasible. If $(\bar{u}, \bar{x}, \bar{z})$ is an optimal solution to (*NLSF($\bar{x}_J$)*), then every $x \in \mathbb{R}^n$ with integer assignment x_J inducing h so that $h = \bar{h}$ as defined in* (4.1) *violates* (4.18).

Proof. In analogy to the proof of Lemma 4.1, we derive the Generalized Benders cut for the problem which is

$$\sum_{\ell \in [k]} \bar{\mu}_\ell \left(h_\ell - \bar{h}_\ell \right) \geq \bar{u}, \tag{4.19}$$

where $\bar{\mu}_\ell$ are the Lagrange multipliers from the KKT system of (NLSF($\bar{x}_J$)).
Suppose that $\bar{h} = h$ and $\sum_{\ell \in [k]} \bar{\mu}_\ell \left(h_\ell - \bar{h}_\ell \right) \geq \bar{u}$. is not violated. Then this equates to $0 \geq \bar{u} > 0$ which is a contradiction to the infeasibility of (NLS($\bar{x}_J$)). The assertion follows. □

Lemma 4.3 states that (BSC-F) cuts off the whole equivalence class associated with a given value $\bar{h}$ rather than just a single infeasible binary assignment similar to Corollary 4.2. We refer to (4.19) as the *feasibility version of the binary symmetric cut*

$$\sum_{\ell \in [k]} \bar{\mu}_\ell \left(h_\ell - \bar{h}_\ell \right) \geq \bar{u}. \tag{BSC-F}$$

It is again sufficient to add (BSC-F) to the outer approximation.

Corollary 4.4. *Let $\bar{x}_J$ be an integer assignment for which (*NLS($\bar{x}_J$)*) is infeasible. If $(\bar{u}, \bar{x}, \bar{z})$ is an optimal solution to (*NLSF($\bar{x}_J$)*), then every $x \in \mathbb{R}^n$ with integer assignment x_J inducing h so that $h = \bar{h}$ violates (*BSC-F*).*

We would like to point out that both cuts, (BSC-O) and (BSC-F), do not involve any constraints that act solely on continuous variables, therefore adding the constraints (2.4) does not affect the formulation of the cuts.

4.3. **An alternative derivation.**

The binary symmetric cuts (BSC-O) and (BSC-F) can also be derived in an alternative fashion.

Given an integral assignment $\bar{x}_J$, (SQE) induces a nonlinear subproblem. Since the integral assignment $\bar{x}_J$ implies an assignment for the h-variables, it suffices to consider the subproblem implied by fixing the h-variables. In this case we replace J, the index set of the integral variables, by the index set of the h-variables as these are the fixed variables in the subproblems. We obtain:

$$\begin{aligned} & \min \quad c_{\bar{J}} x_{\bar{J}} && (4.20)\\ & A_{\bar{J}} x_{\bar{J}} \quad \leq \quad b_{\bar{J}} && (4.21)\\ & \left\| x_{\bar{J}_\ell} \right\|_2^2 \quad \leq \quad h_\ell \text{ for all } \ell \in [k]. && (4.22) \end{aligned}$$

Let us first assume that the resulting subproblem is feasible. The linearizations of the constraints (4.22) in h and $x_{\bar{J}}$ are given by

$$2\bar{x}_{\bar{J}_\ell}^T(x_{\bar{J}_\ell} - \bar{x}_{\bar{J}_\ell}) \leq h_\ell - \bar{h}_\ell \quad \forall \ell \in [k]$$

and the corresponding generalized Benders cut is (BSC-O). Thus, if the problem (SQE) is formulated using the variables h_ℓ, the above linearizations in h can be added without using the binary variables explicitly. In a similar fashion (BSC-F) can be derived.

4.4. **Illustrative example.** We want to elaborate the effect of the strengthened reformulation by the Adams-Sherali substitution in the outer approximation context. Whereas the reformulation (SQE) gives a tighter description of the continuous relaxation of (wSOC), the symmetry exploiting property is due to the linearizations used. On the other hand, outer approximation applied to the original formulation (wSOC) does not yield the same effect as shown in the following example. Consider the following set of constraints

$$x_0 \in \mathbb{R},\ x_1, x_2, x_3 \in \{0,1\},\ x_0^2 + x_1^2 + x_2^2 + x_3^2 \leq 2,\ x_0 \geq 1 \tag{4.23}$$

and its Sherali-Adams based reformulation

$$x_0 \in \mathbb{R},\ x_1, x_2, x_3 \in \{0,1\},\ x_0^2 + x_1 + x_2 + x_3 \leq 2,\ x_0 \geq 1, \tag{4.24}$$

hence we have $J = \{1,2,3\}$, $\bar{J} = \{0\}$ and we define the auxiliary variable corresponding to (4.24) as $h = 2 - (x_1 + x_2 + x_3)$. The binary assignment $\bar{x}_J = (1,1,0)$ with $\bar{h} = 0$ produces infeasible constraints (4.23) and (4.24). Then $(\bar{x}_0, \bar{x}_1, \bar{x}_2, \bar{x}_3) = (1,1,1,0)$ is the solution to the corresponding feasibility problem with objective function value $\bar{u} = 1$. The linearization (4.18) w.r.t. (4.23) is given by

$$2x_0 + 2x_1 + 2x_2 \leq 5 \tag{4.25}$$

and the linearization (4.18) w.r.t. (4.24) is given by

$$2x_0 + x_1 + x_2 + x_3 \leq 3 \tag{4.26}$$

As expected, the assignment $\bar{x}_J = (1,1,0)$ is cut off by both linearizations since both contradict $x_0 \geq 1$. Now consider a different assignment of binary variables $\tilde{x}_J = (0,1,1)$ which corresponds to the same value of $\tilde{h} = \bar{h} = 0$ as $\bar{x}_J$. It hence implies the same infeasible quadratic constraints as $\bar{x}_J$ and is therefore desired to be excluded by the linearizations. We can easily see that $\tilde{x}_J$ is not cut off by the first linearization (4.25), since $\tilde{x}_0 = 1$ provides a feasible point $(1,0,1,1)$. But it is indeed excluded by the second linearization (4.26) as it again contradicts $x_0 \geq 1$. Thus, only (4.26) has the symmetry-breaking effect that was proved in Lemma 4.3.

4.5. **The outer approximation algorithm.** We are ready to formulate an outer approximation algorithm for solving (wSOC) using (BSC-O) and (BSC-F). It is important to keep in mind that the cuts (BSC-O) and (BSC-F), as well as the proposed outer approximation algorithm are defined for the strong quadratic equivalent (SQE).

As usual, throughout the algorithm we maintain the invariant that the (linear) outer approximation problem approximates the feasible region of (SQE) from the outside. We will iteratively add linear constraints to successively tighten the outer approximation of the feasible region of strong quadratic equivalent. Consider the strong quadratic equivalent (SQE) of (wSOC). Then

$$\text{(SQE) without constraints (3.11)} \tag{MIP-OA}$$

is a mixed-integer programming relaxation of (SQE) that results from omitting the conic constraints. Clearly, the feasible region of (MIP-OA) contains the feasible region of (SQE) and so for every optimal solution x_M to (MIP-OA) and every feasible solution x_{SQE} to (SQE), we have

$$c^T x_M \leq c^T x_{SQE}. \tag{4.27}$$

Indeed, if the latter inequality is satisfied with equality for some feasible solution x_{SQE} of (SQE), then this solution must be optimal for (SQE). Clearly, (MIP-OA) can contain additional valid linear inequalities.

Let (MIP-OA$^{(\xi)}$) denote the outer approximation problem (MIP-OA) after ξ rounds of linearizations and further let $x_M^{(\xi)}$ denote an optimal solution to (MIP-OA$^{(\xi)}$). Given an integral assignment obtained from a solution $x_M^{(\xi)}$, we consider the nonlinear subproblems (NLS($x_J^{(\xi)}$)) and (NLSF($x_J^{(\xi)}$)) as introduced in Section 4.2.1 and 4.2.2. For the sake of brevity we define (NLS$^{(\xi)}$) and (NLSF$^{(\xi)}$) respectively.

Let $x_{\text{NLS}}^{(\xi)}$ denote an optimal solution to (NLS$^{(\xi)}$) provided it is feasible and let $u^{(\xi)}$ denote the minimal violation obtained by solving (NLSF$^{(\xi)}$) otherwise. In round ξ of the linearization phase, depending on whether (NLS$^{(\xi)}$) is feasible or not, we add one of the following *binary symmetric cuts*

$$(\text{BSC-O}^{(\xi)}) \qquad \sum_{\ell\in[k]} \mu_\ell^{(\xi)}\left(h_\ell - h_\ell^{(\xi)}\right) \geq c_{\bar{J}}^T (x_{\text{NLS}}^{(\xi)})_{\bar{J}} - c_{\bar{J}}^T x_{\bar{J}} \qquad \text{if (NLS}^{(\xi)}\text{) is feasible,}$$

$$(\text{BSC-F}^{(\xi)}) \qquad \sum_{\ell\in[k]} \mu_\ell^{(\xi)}\left(h_\ell - h_\ell^{(\xi)}\right) \geq u^{(\xi)} \qquad \text{if (NLS}^{(\xi)}\text{) is infeasible}$$

to the outer approximation (MIP-OA) where $\mu_\ell^{(\xi)}$ and $h_\ell^{(\xi)}$ correspond to the respective programs. With (BSC$^{(l)}$) denoting the cut that has been added in round l, the outer approximation program that we consider in round ξ is given by

$$(\text{MIP-OA}^{(\xi)}) = (\text{MIP-OA}) \cap \{(\text{BSC}^{(l)}),\ l \in [\xi-1]\}.$$

We can formulate the following outer approximation algorithm for (SQE).

Algorithm 4.5 (Outer Approximation Algorithm for (SQE))**.**
Input data: Problem of the form (SQE).
Initialization: $\xi \leftarrow 0$, $CUB \leftarrow \infty$, $CLB \leftarrow -\infty$.

```
(1) Solve continuous relaxation of (SQE).
    if (infeasible) return '(SQE) infeasible'
    else            optimal solution $\bar{x}$
                    if   $\bar{x}_J$ binary: return optimal solution $x^* \leftarrow \bar{x}$ with objective value $c^T x^*$.
                    else set up initial outer approximation (MIP-OA$^{(0)}$)
(2) while ($CUB - CLB > eps$)
    (a) Solve (MIP-OA$^{(\xi)}$).
        if (infeasible) return '(SQE) infeasible'
        else            $CLB \leftarrow \max\{CLB, c^T x_M^{(\xi)}\}$
    (b) Solve (NLS$^{(\xi)}$).
        if (infeasible) Solve (NLSF$^{(\xi)}$),
                        (MIP-OA$^{(\xi+1)}$) $\leftarrow$ (MIP-OA$^{(\xi)}$) $\cap$ (BSC-F$^{(\xi)}$)
        else            $CUB \leftarrow \min\{CUB, c^T x_{\text{NLS}}^{(\xi)}\}$,
                        (MIP-OA$^{(\xi+1)}$) $\leftarrow$ (MIP-OA$^{(\xi)}$) $\cap$ (BSC-O$^{(\xi)}$)
    (c) Update $\xi \leftarrow \xi + 1$.
```

We will finally show that Algorithm 4.5 indeed solves(SQE).

Theorem 4.6. *Assume that the feasible region of the continuous relaxation of (SQE) is bounded and that every subproblem (*NLS$^{(\xi)}$*) satisfies a constraint qualification. Then the outer approximation algorithm terminates in a finite number of steps at an optimal solution of (SQE) or with the indication that it is infeasible. Thereby each equivalence class of binary assignments x_J inducing the same values of h is visited at most once.*

Proof. First, note that every feasibility problem satisfies the Slater constraint qualification. Since we furthermore assume a valid constraint qualification for every subproblem (NLS$^{(\xi)}$) of (SQE), the boundedness assumption guarantees optimal solutions that satisfy the KKT optimality conditions. Thus, the binary symmetric cuts can be stated using the Lagrange multipliers satisfying these KKT systems.

It remains to show convergence. Assume (NLS($x_J^{(\xi)}$)) is feasible. Then Lemma 4.1 states that every solution (x, h) with $h = h^{(\xi)}$ that satisfies the binary symmetric cut (BSC-O$^{(\xi)}$) has an objective value greater or equal than $c^T x^{(\xi)}$. Since $x^{(\xi)}$ was an NLP feasible solution it holds that $CUB \leq c^T x^{(\xi)}$. If a solution $(\bar{h}, \bar{x})$ of the outer approximation problem including condition (BSC-O$^{(\xi)}$) satisfies $\bar{h} = h^{(\xi)}$, then $c^T \bar{x} \geq c^T x^{(\xi)}$ must hold. Since $c^T \bar{x} \leq CLB$, the algorithm terminates due to $CUB \leq CLB$.

Otherwise, if (NLS($x^{(\xi)}$)) is infeasible, Lemma 4.3 states that (x, h) satisfying $h = h^{(\xi)}$ is separated by the binary symmetric cut (BSC-F$^{(\xi)}$). Thus, the updated outer approximation cannot result in a new solution with an integer assignment with $h = h^{(\xi)}$. Since the number of possible assignments is finite, convergence of the algorithm follows. ☐

A few remarks are in order.

(1) Convergence of the method is a classical result (cf., e.g., [12]) and is also guaranteed, if the linearizations are based on the original nonlinear constraints in (wSOC). The exclusion of a whole symmetry class corresponding to one binary solution is only valid for cuts based on the strong reformulation (SQE) as illustrated in Subsection 4.4.

(2) The result also holds for variants of the outer approximation algorithm like LP/NLP based branch-and-bound, cf. [18].

(3) It might not hold that a constraint qualification is satisfied for all subproblems. However, it is indeed satisfied by the application problems considered in Section 5. In general a Slater constraint qualification is satisfied for all subproblems whenever the right hand side of (2.2) is strictly positive. If fixing binary variables forces some values h_ℓ for $\ell \in [k]$ to zero, we can eliminate the corresponding variables $x_{\bar{J}_i}$. Assuming $h_\ell > 0$ for the remaining $\ell \in [k]$, $(h_\ell, x_{\bar{J}_\ell})$ for $x_{\bar{J}_\ell} = 0$ provides a strictly feasible point.

(4) Problems (NLS$^{(\xi)}$) or (NLSF$^{(\xi)}$) can be formulated as SOCPs which is indeed relevant if additional conic constraints (2.4) on the continuous variables are present. In that case, the Lagrange multipliers can be identified using the dual solutions of these SOCP representations, for details compare, e.g., [11].

5. Pooling of securities — an application

In the following we consider the problem of optimal pooling of securities in the presence of indivisibilities. Suppose we have a set of pools J and we would to assign securities to this pool. We differentiate between two types of securities: First, we have securities that we can divide into smaller ones so that we can assign fractions of the securities to pools. We call these securities *divisible* and by I_f we denote the corresponding index set. We also have securities that we cannot further divide and that have to be assigned to a pool as a whole. These securities are called *indivisible* and we denote their index set by I_b.

These pools are to be sold of (by securitization or other means) and we would like to pool as many securities as possible without exceeding a certain risk level assigned to each pool (here measured by the variance of the resulting portfolio). The situation is depicted in Figure 5.1. We obtain the optimization problem depicted in Figure 5.2, where $Q^{1/2}$ is the root of a positive-definite covariance matrix of the divisible securities and $C^{1/2}$ is the root of a positive-definite covariance matrix of the indivisible securities. Further, by σ_j with $j \in J$ we denote the accepted risk levels. The risk of each pool is then effectively limited in constraint (5.6). Note that we assume that there is no correlation between divisible and indivisible securities. We also allow for arbitrary additional (combinatorial) constraints

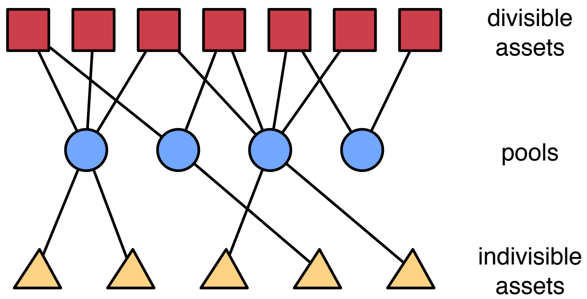

FIGURE 5.1. Pooling of assets under risk constraints

on the (in-)divisible securities respectively (constraints (5.4) and (5.5)). In order to make the pools attractive for investors, the objective function (see (5.1)) asks for a solution that maximizes the return weighted number of assets in each pool; the corresponding return vectors are denoted by r and t.

$$\max_{x,y} \quad \sum_{j\in J}\left(\sum_{i\in I_b} r_i x_{ij}\right) + \left(\sum_{i\in I_f} t_i y_{ij}\right) \tag{5.1}$$

$$\sum_{j\in j} y_{ij} \leq 1 \quad \forall i \in I_f \tag{5.2}$$

$$\sum_{j\in j} x_{ij} \leq 1 \quad \forall i \in I_b \tag{5.3}$$

$$A_f y \leq \delta_f \tag{5.4}$$

$$A_b x \leq \delta_b \tag{5.5}$$

$$\left\|(Q^{1/2}y^j; C^{1/2}x^j)\right\|_2 \leq \sigma_j \quad \forall j \in J \tag{5.6}$$

$$x_{ij} \in \{0,1\} \quad \forall i \in I_b, j \in J \tag{5.7}$$

$$y_{ij} \geq 0 \quad \forall i \in I_f, j \in J \tag{5.8}$$

FIGURE 5.2. Optimal pooling with indivisibilities

6. COMPUTATIONAL RESULTS

We will now present computational results for the Outer Approximation Algorithm 4.5 (BSC) using the symmetry exploiting cuts presented in this paper. We compare the implementation to a classical outer approximation algorithm using subgradient cuts and to the MOSEK® Version 7.0.0.85 solver for mixed integer conic problems applied to the strong reformulation (SQE). We use randomly generated test instances of the form of the pooling problem described in Section 5.

In our comparison, the first algorithm is (MOSEK) with the default options. It uses a branch-and-cut approach to solve the strong reformulation (SQE) as a mixed integer second order cone program. The second algorithm (Classic) uses a linear outer approximation method based on standard subgradient based linearizations (4.3) of the original nonlinear constraints in (wSOC), for details compare [11]. The third method (BSC) is an outer approximation method using the BSC-cuts (BSC-O) and (BSC-F) presented in this paper.

Algorithms (Classic) and (BSC) had been implemented with MATLAB®, release 2013b, where MOSEK® is used as a MIP solver for the outer approximations as well as a conic solver for the occurring SOCP subproblems.

(MOSEK) obj	(Classic) obj	(BSC) obj	(MOSEK) time	(Classic) time	(BSC) time	(MOSEK) nodes,nlps	(Classic) mips,nlps	(BSC) mips,nlps	(Classic) rel.gap
-32.13	-32.13	-32.12	5.44	15.16	0.34	521/510	28/27	3/2	0.01
-33.38	-33.38	-33.38	7.96	13.18	0.16	711/711	35/34	3/2	0.00
-43.37	-43.37	-43.36	23.32	72.18	1.29	2192/2184	30/27	3/2	0.01
-31.69	-31.68	-31.68	1.17	0.31	0.36	77/65	2/1	3/2	0.00
-30.39	-30.38	-30.39	0.50	41.86	0.20	29/28	198/100	7/5	0.00
-30.95	-30.95	-30.95	8.10	33.24	1.17	554/554	28/27	4/3	0.01
-26.85	-26.85	-26.85	3.34	20.22	0.39	285/282	28/27	3/2	0.01
-35.80	-35.80	-35.80	2.28	74.69	1.15	159/153	75/46	8/5	0.01
-29.59	-29.57	-29.59	0.27	34.49	0.19	1/1	192/100	4/2	0.00
-30.03	-30.03	-30.03	7.77	4.84	0.11	683/674	24/23	3/2	0.00

TABLE 1. Computational results (180 variables, 60 binaries, 120 linear constraints)

(MOSEK) obj	(Classic) obj	(BSC) obj	(MOSEK) time	(Classic) time	(BSC) time	(MOSEK) nodes,nlps	(Classic) mips,nlps	(BSC) mips,nlps	(Classic) rel.gap
-10.34	-10.34	-10.34	5.27	107.33	0.51	573/565	84/72	5/4	0.01
-8.87	-8.87	-8.87	4.82	37.39	0.34	539/498	35/34	4/3	0.05
-7.85	-7.85	-7.85	3.00	131.46	0.27	261/261	90/89	3/2	0.03
-7.96	-7.96	-7.96	0.97	38.13	0.17	95/80	158/100	8/5	0.00
-9.30	-9.30	-9.30	1.28	339.82	0.44	130/116	129/100	4/3	0.04
-9.69	-9.68	-9.69	10.34	900.00	1.70	1053/1053	183/99	7/4	0.05
-13.72	-13.72	-13.72	5.79	630.09	1.20	603/602	177/100	8/5	0.02
-9.80	-9.80	-9.80	0.58	90.76	0.44	45/38	133/100	7/5	0.04
-10.94	-10.94	-10.94	2.53	120.98	0.90	247/247	93/49	6/4	0.03
-11.58	-11.58	-11.58	2.95	41.23	0.69	233/226	55/42	7/4	0.02

TABLE 2. Computational results (210 variables, 60 binaries, 95 linear constraints)

We tested the algorithms on randomly generated instances of four different sizes for the application described in Section 5. Tables 1 to 4 show the results for ten instances per problem size. We use a limit of 900 seconds and also limit the number of outer approximation iterations to 100. Tables 1 to 4 display the best objective function values and the running times in seconds. Furthermore, for (MOSEK) we report the number of branch-and-bound nodes (nodes) and SOCP relaxations (rels). For (Classic) and (BSC), we report the number of linear mixed integer subproblems (mips) and SOCP subproblems (nlps) solved by each algorithm, where feasibility problems are counted in addition. The classical outer approximation (Classic) was often stopped due to numerical instabilities or because the time or iteration limit had been reached. In those cases, we also report the resulting integrality gap. For those few test problems, where also (MOSEK) reached the time limit, the integrality gap is not reported, since it was below 1%.

(MOSEK) obj	(Classic) obj	(BSC) obj	(MOSEK) time	(Classic) time	(BSC) time	(MOSEK) nodes,nlps	(Classic) mips,nlps	(BSC) mips,nlps	(Classic) rel.gap
-9.86	-9.84	-9.86	17.07	900.00	4.40	1160/1160	98/54	7/5	0.03
-11.01	-11.01	-11.00	86.58	684.91	11.97	6979/6964	35/34	4/3	0.05
-11.53	-11.52	-11.53	136.97	900.00	21.09	11815/11812	38/37	5/4	0.05
-12.78	-12.78	-12.78	81.15	900.00	20.79	6405/6381	64/36	8/5	0.05
-11.69	-11.69	-11.68	71.01	900.00	11.06	5717/5713	53/52	5/4	0.05
-10.74	-10.65	-10.74	6.93	900.00	6.90	429/422	102/51	12/7	0.09
-13.05	-13.05	-13.05	9.52	900.00	3.01	711/683	118/63	7/4	0.02
-11.89	-11.89	-11.89	71.28	900.00	12.17	5899/5887	64/36	8/5	0.06
-14.22	-14.22	-14.22	68.02	519.47	12.04	5099/5087	40/33	7/4	0.04
-12.83	-12.83	-12.83	62.62	384.93	7.61	5083/5081	35/34	4/3	0.03

TABLE 3. Computational results (280 variables, 105 binaries, 110 linear constraints)

(MOSEK) obj	(Classic) obj	(BSC) obj	(MOSEK) time	(Classic) time	(BSC) time	(MOSEK) nodes,nlps	(Classic) mips,nlps	(BSC) mips,nlps	(Classic) rel.gap
-11.73	-11.69	-11.73	900.00	237.31	116.58	40291/40251	3/2	4/3	0.07
-14.11	-14.10	-14.11	835.26	900.00	82.71	29751/29688	11/10	4/3	0.03
-16.26	-16.26	-16.26	564.57	900.00	35.29	18859/18823	30/29	4/3	0.06
-14.01	-13.99	-14.01	654.19	900.00	193.18	23861/23826	7/6	4/3	0.05
-13.69	-13.65	-13.69	551.81	900.00	138.08	19919/19843	13/8	6/4	0.08
-17.03	-17.02	-17.03	900.00	900.00	382.97	42103/41979	4/3	4/3	0.07
-14.23	-14.23	-14.23	585.61	900.00	27.42	19891/19882	20/19	4/3	0.03
-16.78	-16.77	-16.77	645.66	900.00	44.69	23767/23751	25/24	6/5	0.05
-12.93	-12.88	-12.93	814.50	900.00	83.82	31382/31303	3/2	4/3	0.07
-11.73	-11.69	-11.73	900.00	218.04	110.15	40291/40251	3/2	4/3	0.07

TABLE 4. Computational results (480 variables, 160 binaries, 180 linear constraints)

The problems in Tables 1, 2 and 3 could be solved to optimality by (MOSEK) as well as (BSC), where (Classic) could not close the gap for several (even small) instances.

For the large problems in Table 4, also (MOSEK) reached the time limit of 900 seconds in three cases, whereas (BSC) could still solve all problems to optimality in less than 400 seconds.

We observe that in all cases the outer approximation algorithm with binary symmetric cuts (BSC) requires to solve a much smaller amount of MIPs and SOCPs than the classical outer approximation algorithm (Classic) resulting in significantly shorter running times. This highlights the symmetry-breaking effect of the cuts applied. Also, the number of SOCP problems (nlps) solved by (BSC) is significantly smaller in comparison to the SOCP relaxations (rels) solved by (MOSEK); we have a similar situation for the mixed-integer programs (mips). This is also clearly reflected in the running times of (BSC), which are vastly shorter than the running times of (MOSEK), especially for the larger problems. It is important to note, that (MOSEK) already solves the strong reformulation, giving a significant advantage over the standard formulation. The better performance of (BSC) in particular highlights the effectiveness of the cuts exploiting the symmetry in the problem.

7. Final remarks

We presented cutting planes for a certain class of mixed-0/1 second-order cone programs where the coupling of continuous and binary variables occurs only in the conic constraints (see 2.1). These cutting-planes exploit symmetries arising from this special type of coupling and can lead to significant computational savings. It is straight forward to generalize (wSOC) to classes where only a subset of the binaries occur solely in conic constraints. In this case the cuts (BSC-O) and (BSC-F) contain *h* variables for the subset of binaries occurring only in conic constraints and another set of variables and Lagrange multipliers for other binary variables. A typical example for such a setting could be an additional capital constraint for the application problem sketched in Section 5. The cuts (BSC-O) and (BSC-F) would separate whole equivalence classes for a *given* amount of capital then. In the presence of few additional coupling constraints the symmetry exploiting effect of these cuts can still be considerable.

Acknowledgements

The authors are indebted to the two anonymous referees for the helpful comments. Research partially supported by German Research Foundation (DFG) funded SFB 805 and by NSF grant CMMI-1333789.

References

[1] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, 95(1):3–51, 2003.

[2] A. Atamt rk and V. Narayanan. Cuts for Conic Mixed-Integer Programming. In *Integer programming and combinatorial optimization: 12th International IPCO Conference, Ithaca, NY, USA, June 25-27, 2007: proceedings*, page 16. Springer-Verlag New York Inc, 2007.

[3] A. Atamt rk and V. Narayanan. Lifting for conic mixed-integer programming. *Mathematical Programming*, pages 1–13, 2010.

[4] J.F. Benders. Partitioning for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[5] P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuejols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. W chter. An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs. Technical report, IBM Research Division, New York, 2005, 2005.

[6] P. Bonami, M. Kilinc, and J. Linderoth. Algorithms and Software for Convex Mixed Integer Nonlinear Programs. Technical report, Computer Sciences Department, University of Wisconsin-Madison, 2009, 2009.

[7] M.T. ezik and G. Iyengar. Cuts for Mixed 0-1 Conic Programming. *Math.Programming, Ser. A*, 2005.

[8] M. Charikar, K. Makarychev, and Y. Makarychev. Integrality gaps for Sherali-Adams relaxations. *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 283–292, 2009.

[9] Y. Chu and Q. Xia. Generating Benders Cuts for a General Class of Integer Programming Problems. *Lecture Notes in Computer Science*, 2004.

[10] S. Drewes. *Mixed Integer Second Order Cone Programming*. PhD thesis, Technische Universit t Darmstadt, 2009.

[11] S. Drewes and U. Ulbrich. Subgradient based Outer Approximation for Mixed Integer Second Order Cone Programming. In *IMA Volume Series, Springer Special Issue on Mixed-Integer Nonlinear Optimization: Algorithmic Advances and Applications*. SPRINGER, to appear 2011.

[12] R. Fletcher and S. Leyffer. Solving Mixed Integer Nonlinear Programs by Outer Approximation. *Mathematical Programming*, 66:327–349, 1994.

[13] A.M. Geoffrion. Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, 10:237–260, 1972.

[14] I.E. Grossmann and Z. Kravanja. Mixed-integer nonlinear programming: a survey of algorithms and applications. In *Large-Scale Optimization with Applications, Part II: Optimal Design and Control*. 1997.

[15] M. Laurent. A comparison of the Sherali-Adams, Lov sz-Schrijver, and Lasserre relaxations for 0-1 programming. *Mathematics of Operations Research*, 28:470–496, 2003.

[16] C. Mathieu and A. Sinclair. Sherali-Adams relaxations of the matching polytope. *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 293–302, 2009.

[17] S. Pokutta and A.S. Schulz. On the connection of the Sherali-Adams closure and border bases. *submitted / preprint available at* `http://www.optimization-online.org/DB_HTML/2009/08/2378.html`, 2009.

[18] I. Quesada and I.E. Grossmann. An LP/NLP based Branch and Bound Algorithm for Convex MINLP Optimization Problems. *Computers and Chemical Engineering*, 16(10,11):937–947, 1992.

[19] H.D. Sherali and W.P. Adams. A hierarchy of relaxations between the continous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3:411–430, 1990.

[20] H.D. Sherali and B.M.P. Fraticelli. A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22(1):319–342, 2002.

[21] R.A. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86(3):515–532, 1999.

[22] J.P. Vielma, S. Ahmed, and G.L. Nemhauser. A Lifted Linear Programming Branch-and-Bound Algorithm for Mixed-Integer Conic Quadratic Programs. *INFORMS Journal on Computing*, 20(3):438–450, 2008.

MathWorks Consulting Services, Germany
E-mail address: `Sarah.Drewes@mathworks.de`

ISyE, Georgia Institute of Technology, Atlanta, USA
E-mail address: `sebastian.pokutta@isye.gatech.edu`