SUBGRADIENT BASED OUTER APPROXIMATION FOR MIXED INTEGER SECOND ORDER CONE PROGRAMMING

SARAH DREWES* AND STEFAN ULBRICH †

Abstract. This paper deals with outer approximation based approaches to solve mixed integer second order cone programs. Thereby the outer approximation is based on subgradients of the second order cone constraints. Using strong duality of the subproblems that are solved during the algorithm, we are able to determine subgradients satisfying the KKT optimality conditions. This enables us to extend convergence results valid for continuously differentiable mixed integer nonlinear problems to subdifferentiable constraint functions. Furthermore, we present a version of the branch-and-bound based outer approximation that converges when relaxing the convergence assumption that every SOCP satisfies the Slater constraint qualification. We give numerical results for some application problems showing the performance of our approach.

Key words. Mixed Integer Nonlinear Programming, Second Order Cone Programming, Outer Approximation

AMS(MOS) subject classifications. 90C11

1. Introduction.

Mixed Integer Second Order Cone Programs (MISOCP) can be formulated as

$$\begin{array}{rrcll} \min & c^T x & & & \\ s.t. & Ax & = & b & \\ & x & \succeq & 0 & \\ & (x)_j & \in & [l_j, u_j] & (j \in J), \\ & (x)_j & \in & \mathbb{Z} & (j \in J), \end{array} \tag{1.1}$$

where $c = (c_1^T, \dots c_{noc}^T)^T \in \mathbb{R}^n$, $A = (A_1, \dots A_{noc}) \in \mathbb{R}^{m,n}$, with $c_i \in \mathbb{R}^{k_i}$ and $A_i \in \mathbb{R}^{m,k_i}$ for $i \in \{1, \dots noc\}$ and $\sum_{i=1}^{noc} k_i = n$. Furthermore, $b \in \mathbb{R}^m$, $(x)_j$ denotes the j-th component of x, $l_j, u_j \in \mathbb{R}$ for $j \in J$ and $J \subset \{1, \dots n\}$ denotes the integer index set. Here, $x \succeq 0$ for $x = (x_1^T, \dots x_{noc}^T)^T$ with $x_i \in \mathbb{R}^{k_i}$ for $i \in \{1, \dots noc\}$ denotes that

$$x \in \mathcal{K},\ \mathcal{K} := \mathcal{K}_1 \times \dots \times \mathcal{K}_{noc},$$

where

$$\mathcal{K}_i := \{x_i = (x_{i0}, x_{i1}^T)^T \in \mathbb{R} \times \mathbb{R}^{k_i - 1} : \|x_{i1}\|_2 \le x_{i0}\}$$

is the second order cone of dimension k_i. Mixed integer second order cone problems have various applications in finance or engineering, for example

*Research Group Nonlinear Optimization, Department of Mathematics, Technische Universität Darmstadt, Germany.

†Research Group Nonlinear Optimization, Department of Mathematics, Technische Universität Darmstadt, Germany.

turbine balancing problems, cardinality-constrained portfolio optimization (cf. Bertsimas and Shioda in [18] or Vielma et al. in [11]) or the problem of finding a minimum length connection network also known as the Euclidean Steiner Tree Problem (ESTP) (cf. Fampa, Maculan in [16]).

Available convex MINLP solvers like `BONMIN` [23] by Bonami et al. or `FilMINT` [26] by Abhishek et al. are in general not applicable for (1.1), since the occurring second order cone constraints are not continuously differentiable.

Branch-and-cut methods for convex mixed 0-1 problems have been discussed by Stubbs and Mehrotra in [2] and [10] which can be applied to solve (1.1), if the integer variables are binary. In [5] Çezik and Iyengar discuss cuts for general self-dual conic programming problems and investigate their applications on the maxcut and the traveling salesman problem. Atamtürk and Narayanan present in [13] integer rounding cuts for conic mixed-integer programming by investigating polyhedral decompositions of the second order cone conditions and in [12] the authors discuss lifting for mixed integer conic programming, where valid inequalities for mixed-integer feasible sets are derived from suitable subsets.
One article dealing with non-differentiable functions in the context of outer approximation approaches for MINLP is [7] by Fletcher and Leyffer, where the authors prove convergence of outer approximation algorithms for non-smooth penalty functions. The only article dealing with outer approximation techniques for MISOCPs is [11] by Vielma et al., which is based on Ben-Tal and Nemirovskii's polyhedral approximation of the second order cone constraints [14]. Thereby, the size of the outer approximation grows when strengthening the precision of the approximation. This precision and thus the entire outer approximation is chosen in advance, whereas the approximation presented here is strengthened iteratively in order to guarantee convergence of the algorithm.
In this paper we present a hybrid branch-and-bound based outer approximation approach for MISOCPs. The approach is based on the branch-and-bound based outer approximation approach for continuously differentiable constraints – as proposed by Bonami et al. in [9] on the basis of Fletcher and Leyffer [7] and Quesada and Grossmann [3]. The idea is to iteratively compute integer feasible solutions of a (sub)gradient based linear outer approximation of (1.1) and to tighten this outer approximation by solving nonlinear continuous problems.
Thereby linear outer approximations based on subgradients satisfying the Karush Kuhn Tucker (KKT) optimality conditions of the occurring SOCP problems enable us to extend the convergence result for continuously differentiable constraints to subdifferentiable second order cone constraints.
Thus, in contrast to [11], the subgradient based approximation induces convergence of any classical outer approximation based approach under

the specified assumptions. We also present an adaption of the algorithm that converges even if one of these convergence assumptions is violated. In numerical experiments we show the applicability of the algorithm and compare it to a nonlinear branch-and-bound approach.

2. Preliminaries.

In the following $int(\mathcal{K}_i)$ denotes the interior of the cone $\mathcal{K}_i$, i.e. those vectors x_i satisfying $x_{i0} > \|x_{i1}\|$, $bd(\mathcal{K}_i)$ denotes the boundary of $\mathcal{K}_i$, i.e. those vectors x_i satisfying $x_{i0} = \|x_{i1}\|$. By $\|\cdot\|$ we denote the Euclidean norm. Assume $g : \mathbb{R}^n \mapsto \mathbb{R}$ is a convex and subdifferentiable function on $\mathbb{R}^n$. Then due to the convexity of g, the inequality $g(x) \geq g(\bar{x}) + \xi^T(x-\bar{x})$ holds for all $\bar{x}, x \in \mathbb{R}^n$ and every subgradient $\xi \in \partial g(\bar{x})$ – see for example [20]. Thus, we obtain a linear outer approximation of the region $\{x : g(x) \leq 0\}$ applying constraints of the form

$$g(\bar{x}) + \xi^T(x - \bar{x}) \leq 0. \tag{2.1}$$

In the case of (1.1), the feasible region is described by constraints

$$g_i(x) := -x_{i0} + \|x_{i1}\| \leq 0,\ i = 1, \dots noc, \tag{2.2}$$

where $g_i(x)$ is differentiable on $\mathbb{R}^n \setminus \{x : \|x_{i1}\| = 0\}$ with $\nabla g_i(x_i) = (-1, \frac{x_{i1}^T}{\|x_{i1}\|})$ and subdifferentiable if $\|x_{i1}\| = 0$. The following lemma gives a detailed description of the subgradients of (2.2).

LEMMA 2.1. *The convex function $g_i(x_i) := -x_{i0} + \|x_{i1}\|$ is subdifferential in $x_i = (x_{i0}, x_{i1}^T)^T = (a, 0^T)^T$, $a \in \mathbb{R}$, with $\partial g_i((a, 0^T)^T) = \{\xi = (\xi_0, \xi_1^T)^T, \xi_0 \in \mathbb{R}, \xi_1 \in \mathbb{R}^{k_i - 1} : \xi_0 = -1, \|\xi_1\| \leq 1\}$.*

Proof. Follows from the subgradient inequality in $(a, 0^T)^T$. □

The following lemma investigates a complementary constraint on two elements of the second order cone that is used in the subsequent sections.

LEMMA 2.2. *Assume $\mathcal{K}$ is the second order cone of dimension k and $x = (x_0, x_1^T)^T \in \mathcal{K}, s = (s_0, s_1^T)^T \in \mathcal{K}$ satisfy the condition $x^T s = 0$, then*

1. *$x \in int(\mathcal{K}) \Rightarrow s = (0, \dots 0)^T$,*
2. *$x \in bd(\mathcal{K}) \setminus \{0\} \Rightarrow s \in bd(\mathcal{K})$ and $\exists \gamma \geq 0 :\ s = \gamma(x_0, -x_1^T)$.*

Proof. 1.: Assume $\|x_1\| > 0$ and $s_0 > 0$. Due to $x_0 > \|\bar{x}_1\|$ it holds that $s^T x = s_0 x_0 + s_1^T x_1 > s_0\|x_1\| + s_1^T x_1 \geq s_0\|x_1\| - \|s_1\|\|x_1\|$. Then $x^T s = 0$ can only be true, if $s_0\|x_1\| - \|s_1\|\|x_1\| < 0 \Leftrightarrow s_0 < \|s_1\|$ which contradicts $s \in \mathcal{K}$. Thus, $s_0 = 0 \Rightarrow s = (0, \dots 0)^T$. If $\|x_1\| = 0$, then $s_0 = 0$ follows directly from $x_0 > 0$.
2.: Due to $x_0 = \|x_1\|$, we have $s^T x = 0 \Leftrightarrow -s_1^T x_1 = s_0\|x_1\|$. Since $s_0 \geq \|s_1\| \geq 0$ we have $-s_1^T x_1 = s_0\|x_1\| \geq \|x_1\|\|s_1\|$. Cauchy -Schwarz's inequality yields $-s_1^T x_1 = \|x_1\|\|s_1\|$ which implies both $s_1 = -\gamma x_1,\ \gamma \in \mathbb{R}$ and $s_0 = \|s_1\|$. It follows that $-x_1^T s_1 = \gamma x_1^T x_1 \geq 0$. Together with $s_0 = \|s_1\|$ and $\|x_1\| = x_0$ we get that there exists $\gamma \geq 0$, such that $s_1 = (\|-\gamma x_1\|, -\gamma x_1^T)^T = \gamma(x_0, -x_1^T)^T$. □

We make the following assumptions:

A1 The set $\{x : Ax = b, x_J \in [l, u]\}$ is bounded.
A2 Every nonlinear subproblem $(NLP(x_J^k))$ that is obtained from (1.1) by fixing the integer variables to the value x_J^k has nonempty interior (Slater constraint qualification).

These assumptions comply with the assumptions made by Fletcher and Leyffer in [7] as follows. We drop the assumption of continuous differentiability of the constraint functions, but we assume a constraint qualification inducing strong duality instead of an arbitrary constraint qualification which suffices in the differentiable case.

Remark. A2 might be expected as a strong assumption, since it is violated as soon as a leading cone variable x_{i0} is fixed to zero. In that case, all variables belonging to that cone can be eliminated and the Slater condition may hold now for the reduced problem. Moreover, at the end of Section 5, we present an enhancement of the algorithm that converges even if assumption A2 is violated.

3. Feasible Nonlinear Subproblems. For a given integer configuration x_J^k, we define the SOCP subproblem

$$\begin{array}{lrl} \min & c^T x & \\ s.t. & Ax & = b, \\ & x & \succeq 0, \\ & x_J & = x_J^k. \end{array} \qquad (NLP(x_J^k))$$

The dual of $(NLP(x_J^k))$, in the sense of Nesterov and Nemirovskii [19] or Alizadeh and Goldfarb [8], is given by

$$\begin{array}{lrl} \max & (b^T, x_J^{k,T})y & \\ s.t. & (A^T, I_J^T)y + s & = c, \\ & s & \succeq 0, \end{array} \qquad (NLP(x_J^k)\text{-D})$$

where $I_J = ((I_J)_1, \ldots (I_J)_{noc})$ denotes the matrix mapping x to the integer variables x_J where $(I_J)_i \in \mathbb{R}^{|J|,k_i}$ is the block of columns of I_J associated with the i-th cone of dimension k_i. We define

$$\begin{array}{lcl} I_0(\bar{x}) & := & \{i : \bar{x}_i = (0, \ldots 0)^T\}, \\ I_a(\bar{x}) & := & \{i : g_i(\bar{x}) = 0, \bar{x}_i \neq (0, \ldots 0)^T\}, \end{array} \qquad (3.1)$$

where $I_a(\bar{x})$ is the index set of active conic constraints that are differentiable in $\bar{x}$ and $I_0(\bar{x})$ is the index set of active constraints that are subdifferentiable in $\bar{x}$. The crucial point in an outer approximation approach is to tighten the outer approximation problem such that the integer assignment of the last solution is cut off. Assume x_J^k is this last solution. Then we will show later that those subgradients in $\partial g_i(\bar{x})$ that satisfy the KKT conditions in the solution $\bar{x}$ of $(NLP(x_J^k))$ give rise to linearizations with this tightening property. Hence, we show now, how to choose elements $\bar{\xi}_i$ in the subdifferentials $\partial g_i(\bar{x})$ for $i \in \{1, \dots noc\}$ that satisfy the KKT conditions

$$\begin{aligned} c_i + (A_i^T, (I_J)_i^T)\bar{\mu} + \bar{\lambda}_i \bar{\xi}_i &= 0, \quad i \in I_0(\bar{x}), \\ c_i + (A_i^T, (I_J)_i^T)\bar{\mu} + \bar{\lambda}_i \nabla g_i(\bar{x}_i) &= 0, \quad i \in I_a(\bar{x}), \\ c_i + (A_i^T, (I_J)_i^T)\bar{\mu} &= 0, \quad i \notin I_0(\bar{x}) \cup I_a(\bar{x}) \end{aligned} \tag{3.2}$$

in the solution $\bar{x}$ of $(NLP(x_J^k))$ with appropriate Lagrange multipliers $\bar{\mu}$ and $\bar{\lambda} \geq 0$. This step is not necessary if the constraint functions are continuously differentiable, since $\partial g_i(\bar{x})$ then contains only one element: the gradient $\nabla g_i(\bar{x})$.

LEMMA 3.1. *Assume A1 and A2. Let $\bar{x}$ solve $(NLP(x_J^k))$ and let $(\bar{s}, \bar{y})$ be the corresponding dual solution of $(NLP(x_J^k)$-D). Then there exist Lagrange multipliers $\bar{\mu} = -\bar{y}$ and $\bar{\lambda}_i \geq 0$ $(i \in I_0 \cup I_a)$ that solve the KKT conditions (3.2) in $\bar{x}$ with subgradients*

$$\bar{\xi}_i = \begin{pmatrix} -1 \\ -\frac{\bar{s}_{i1}}{\bar{s}_{i0}} \end{pmatrix}, \text{ if } \bar{s}_{i0} > 0, \ \bar{\xi}_i = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \text{ if } \bar{s}_{i0} = 0 \qquad (i \in I_0(\bar{x})).$$

Proof. A1 and A2 guarantee the existence of a primal-dual solution $(\bar{x}, \bar{s}, \bar{y})$ satisfying the primal dual optimality system (cf. Alizadeh and Goldfarb [8])

$$c_i - (A_i^T, (I_J)_i^T)\bar{y} = \bar{s}_i, \ i = 1, \dots noc, \tag{3.3}$$
$$A\bar{x} = b, \ I_J \bar{x} = x_J^k, \tag{3.4}$$
$$\bar{x}_{i0} \geq \|\bar{x}_{i1}\|, \ \bar{s}_{i0} \geq \|\bar{s}_{i1}\|, \ i = 1, \dots noc, \tag{3.5}$$
$$\bar{s}_i^T \bar{x}_i = 0, \ i = 1, \dots noc. \tag{3.6}$$

Since $(NLP(x_J^k))$ is convex and due to A2, there also exist Lagrange multipliers $\bar{\mu} \in \mathbb{R}^m$, $\bar{\lambda} \in \mathbb{R}^{noc}$, such that $\bar{x}$ satisfies the KKT-conditions (3.2) with elements $\bar{\xi}_i \in \partial g_i(\bar{x})$. We now compare both optimality systems to each other.

First, we consider $i \notin I_0 \cup I_a$ and thus $\bar{x}_i \in int(\mathcal{K}_i)$. Lemma 2.2, part 1 induces $\bar{s}_i = (0, \dots 0)^T$. Conditions (3.3) for $i \notin I_0 \cup I_a$ are thus equal to $c_i - (A_i^T, (I_J)_i^T)\bar{y} = 0$ and thus $\bar{\mu} = -\bar{y}$ satisfies the KKT-condition (3.2) for $i \notin I_0 \cup I_A$.

Next we consider $i \in I_a(\bar{x})$, where $x_i \in bd(\mathcal{K}) \setminus \{0\}$. Lemma 2.2, part 2 yields

$$\bar{s}_i = \begin{pmatrix} \| - \gamma \bar{x}_{i1}\| \\ -\gamma \bar{x}_{i1} \end{pmatrix} = \gamma \begin{pmatrix} \bar{x}_{i0} \\ -\bar{x}_{i1} \end{pmatrix} \tag{3.7}$$

for $i \in I_a(\bar{x})$. Inserting $\nabla g_i(\bar{x}) = (-1, \frac{\bar{x}_{i1}^T}{\|\bar{x}_{i1}\|})^T$ for $i \in I_a$ into (3.2) yields the existence of $\lambda_i \geq 0$ such that

$$c_i + (A_i^T, (I_J)_i^T)\mu = \lambda_i \begin{pmatrix} 1 \\ -\frac{\bar{x}_{i1}}{\|\bar{x}_{i1}\|} \end{pmatrix}, \ i \in I_a(\bar{x}). \tag{3.8}$$

Insertion of (3.7) into (3.3) and comparison with (3.8) yields the existence of $\gamma \geq 0$ such that $\bar{\mu} = -\bar{y}$ and $\bar{\lambda}_i = \gamma \bar{x}_{i0} = \gamma \|\bar{x}_{i1}\| \geq 0$ satisfy the KKT-conditions (3.2) for $i \in I_a(\bar{x})$.
For $i \in I_0(\bar{x})$, condition (3.2) is satisfied by $\mu \in \mathbb{R}^m$, $\bar{\lambda}_i \geq 0$ and subgradients $\bar{\xi}_i$ of the form $\bar{\xi}_i = (-1, v^T)^T$, $\|v\| \leq 1$. Since $\bar{\mu} = -\bar{y}$ satisfies (3.2) for $i \notin I_0$, we look for a suitable v and $\bar{\lambda}_i \geq 0$ satisfying $c_i - (A_i^T, (I_J)_i^T)\bar{y} = \bar{\lambda}_i(1, -v^T)^T$ for $i \in I_0(\bar{x})$. Comparing the last condition with (3.3) yields that if $\|\bar{s}_{i1}\| > 0$, then $\bar{\lambda}_i = \bar{s}_{i0}, -v = \frac{\bar{s}_{i1}}{\bar{s}_{i0}}$ satisfy condition (3.2) for $i \in I_0(\bar{x})$. Since $\bar{s}_{i0} \geq \|\bar{s}_{i1}\|$ we obviously have $\bar{\lambda}_i \geq 0$ and $\|v\| = \|\frac{\bar{s}_{i1}}{\bar{s}_{i0}}\| = \frac{1}{\bar{s}_{i0}}\|\bar{s}_{i1}\| \leq 1$. If $\|\bar{s}_{i1}\| = 0$, the required condition (3.2) is satisfied by $\bar{\lambda}_i = \bar{s}_{i0}, -v = (0, \dots 0)^T$. □

4. Infeasible Nonlinear Subproblems. If the nonlinear program $(NLP(x_J^k))$ is infeasible for x_J^k, the algorithm solves a feasibility problem of the form

$$
\begin{array}{rcl}
\min\ u & & \\
s.t.\quad Ax & = & b, \\
-x_{i0} + \|x_{i1}\| & \leq & u,\ i = 1, \dots noc, \\
u & \geq & 0, \\
x_J & = & x_J^k.
\end{array}
\qquad (F(x_J^k))
$$

It has the property that the optimal solution $(\bar{x}, \bar{u})$ minimizes the maximal violation of the conic constraints. The dual program of $(F(x_J^k))$ is

$$
\begin{array}{lrcl}
\max & (b^T, x_J^{k,T})y & & \\
s.t. & -(A^T, I_J^T)y + s & = & 0, \\
 & s_u + \sum_{i=1}^{noc} s_{i0} & = & 1, \\
 & \|s_{i1}\| & \leq & s_{i0},\ i = 1, \dots noc, \\
 & s_u & \geq & 0.
\end{array}
\qquad (F(x_J^k)\text{-D})
$$

We define the index sets of active constraints in a solution $(\bar{x}, \bar{u})$ of $(F(x_J^k))$,

$$
\begin{array}{rcl}
I_F := I_F(\bar{x}) & := & \{i \in \{1, \dots noc\} : -\bar{x}_{i0} + \|\bar{x}_{i1}\| = \bar{u}\}, \\
I_{F0} := I_{F0}(\bar{x}) & := & \{i \in I_F : \|\bar{x}_{i1}\| = 0\}, \\
I_{F1} := I_{F1}(\bar{x}) & := & \{i \in I_F : \|\bar{x}_{i1}\| \neq 0\}.
\end{array}
\qquad (4.1)
$$

One necessity for convergence of the outer approximation approach is the following. Analogously to the feasible case, the solution of the feasibility problem $(F(x_J^k))$ must tighten the outer approximation such that the current integer assignment x_J^k is no longer feasible for the linear outer approximation. For this purpose, we identify subgradients $\xi_i \in \partial g_i(\bar{x})$ at the solution $(\bar{u}, \bar{x})$ of $(F(x_J^k))$ that satisfy the KKT conditions of $(F(x_J^k))$

$$A_i^T \mu_A + (I_J)_i^T \mu_J = 0,\ i \notin I_F, \tag{4.2}$$
$$\nabla g_i(\bar{x}_i)\lambda_{g_i} + A_i^T \mu_A + (I_J)_i^T \mu_J = 0,\ i \in I_{F1}, \tag{4.3}$$
$$\xi_i \lambda_{g_i} + A_i^T \mu_A + (I_J)_i^T \mu_J = 0,\ i \in I_{F0}, \tag{4.4}$$
$$\sum_{i \in I_F} (\lambda_g)_i = 1. \tag{4.5}$$

LEMMA 4.1. *Assume A1 and A2 hold. Let $(\bar{x}, \bar{u})$ solve $(F(x_J^k))$ with $\bar{u} > 0$ and let $(\bar{s}, \bar{y})$ be the solution of its dual program $(F(x_J^k)$-$D)$. Then there exist Lagrange multipliers $\bar{\mu} = -\bar{y}$ and $\bar{\lambda}_i \geq 0$ $(i \in I_F)$ that solve the KKT conditions in $(\bar{x}, \bar{u})$ with subgradients*

$$\xi_i = \begin{pmatrix} -1 \\ -\frac{\bar{s}_{i1}}{\bar{s}_{i0}} \end{pmatrix},\ \text{if } \bar{s}_{i0} > 0,\ \xi_i = \begin{pmatrix} -1 \\ 0 \end{pmatrix},\ \text{if } \bar{s}_{i0} = 0 \tag{4.6}$$

for $i \in I_{F0}(\bar{x})$.

Proof. Since $(F(x_J^k))$ has interior points, there exist Lagrange multipliers $\mu \in \mathbb{R}^m$, $\lambda \geq 0$, such that optimal solution $(\bar{x}, \bar{u})$ of $(F(x_J^k))$ satisfies the KKT-conditions (4.2) - (4.5) with $\xi_i \in \partial g_i(\bar{x}_i)$ plus the feasibility conditions. We already used the complementary conditions for $\bar{u} > 0$ and the inactive constraints. Due to the nonempty interior of $(F(x_J^k))$, $(\bar{x}, \bar{u})$ satisfies also the primal-dual optimality system

$$Ax = b,$$
$$u \geq 0,$$
$$-A_i^T y_A - (I_J^T)_i y_J = s_i,\ i = 1, \dots noc, \tag{4.7}$$
$$x_{i0} + u \geq \|\bar{x}_{i1}\|,\ \sum_{i=1}^{noc} s_{i0} = 1, \tag{4.8}$$
$$s_{i0} \geq \|\bar{s}_{i1}\|,\ i = 1, \dots noc, \tag{4.9}$$
$$s_{i0}(x_{i0} + u) + s_{i1}^T x_{i1} = 0,\ i = 1, \dots noc, \tag{4.10}$$

where we again used complementarity for $\bar{u} > 0$.
First we investigate $i \notin I_F$. In this case $\bar{x}_{i0} + \bar{u} > \|\bar{x}_{i1}\|$ induces $s_i = (0, \dots 0)^T$ (cf. Lemma 2.2, part 1). Thus, the KKT conditions (4.2) are satisfied by $\mu_A = -y_A$ and $\mu_J = -y_J$.
Next, we consider $i \in I_{F1}$ for which by definition $\bar{x}_{i0} + \bar{u} = \|\bar{x}_{i1}\| > 0$ holds. Lemma 2.2, part 2 states that there exists $\gamma \geq 0$ with $s_{i0} = \gamma(\bar{x}_{i0} + \bar{u}) = \gamma\|\bar{x}_{i1}\|$ and $s_{i1} = -\gamma \bar{x}_{i1}$. Insertion into (4.7) yields

$$-A_i^T y_A - (I_J)_i y_J + \gamma \|\bar{x}_{i1}\| \begin{pmatrix} -1 \\ \frac{\bar{x}_{i1}}{\|\bar{x}_{i1}\|} \end{pmatrix} = 0,\ i \in I_{F1}.$$

Since $\nabla g_i(\bar{x}_i) = (-1, \frac{\bar{x}_{i1}^T}{\|\bar{x}_{i1}\|})^T$, we obtain that the KKT-condition (4.3) is satisfied by $\mu_A = -y_A$, $\mu_J = -y_J$ and $\lambda_{g_i} = s_{i0} = \gamma\|\bar{x}_{i1}\| \geq 0$.
Finally, we investigate $i \in I_{F0}$, where $\bar{x}_{i0}+\bar{u} = \|\bar{x}_{i1}\| = 0$. Since $\mu_A = -y_A$, $\mu_J = -y_J$ satisfy the KKT-conditions for $i \notin I_{F0}$, we derive a subgradient ξ_i that satisfies (4.4) with that choice. In analogy to Lemma 3.1 from Section 3 we derive that $\xi_i = (-1, \xi_{i1}^T)^T$ with $\xi_{i1} = \frac{-s_{i1}}{s_{i0}}$, if $s_{i0} > 0$ and $\xi_{i1} = 0$ otherwise, are suitable together with $\lambda_i = s_{i0} \geq 0$. □

5. The Algorithm. Let $T \subset \mathbb{R}^n$ contain solutions of nonlinear subproblems $(NLP(x_J^k))$ and let $S \subset \mathbb{R}^n$ contain solutions of feasibility problems $(F(x_J^k))$. We build a linear outer approximation of (1.1) based on subgradient based linearizations of the form (2.1). Thereby we use the subgradients specified in Lemma 3.1 and 4.1. This gives rise to the mixed integer linear outer approximation problem

$$
\begin{array}{rrcl}
 & \min\ c^T x & & \\
s.t. & Ax & = & b \\
 & c^T x & < & c^T\bar{x},\ \bar{x} \in T,\ \bar{x}_J \in \mathbb{Z}^{|J|}, \\
 & -\|\bar{x}_{i1}\|x_{i0} + \bar{x}_{i1}^T x_{i1} & \leq & 0,\ i \in I_a(\bar{x}),\ \bar{x} \in T, \\
 & -\|\bar{x}_{i1}\|x_{i0} + \bar{x}_{i1}^T x_{i1} & \leq & 0,\ i \in I_{F1}(\bar{x})\ \bar{x} \in S, \\
 & -x_{i0} & \leq & 0,\ i \in I_0(\bar{x}),\ \bar{s}_{i0} = 0,\ \bar{x} \in T, \\
 & -x_{i0} - \frac{1}{\bar{s}_{i0}}\bar{s}_{i1}^T x_{i1} & \leq & 0,\ i \in I_0(\bar{x}),\ \bar{s}_{i0} > 0,\ \bar{x} \in T, \\
 & -x_{i0} - \frac{1}{\bar{s}_{i0}}\bar{s}_{i1}^T x_{i1} & \leq & 0,\ i \in I_{F0}(\bar{x}), \bar{s}_{i0} > 0,\ \bar{x} \in S, \\
 & -x_{i0} & \leq & 0,\ i \in I_{F0}(\bar{x}), \bar{s}_{i0} = 0,\ \bar{x} \in S, \\
 & x_j & \in & [l_j, u_j],\ (j \in J) \\
 & x_j & \in & \mathbb{Z},\ (j \in J).
\end{array}
\qquad \text{(MIP(T,S))}
$$

The idea of outer approximation based algorithms is to use such a linear outer approximation (MIP(T,S))of the original problem (1.1) to produce integer assignments. For each integer assignment the nonlinear subproblem $(NLP(x_J^k))$ is solved generating feasible solutions for (1.1) as long as $(NLP(x_J^k))$ is feasible. We define nodes N^k consisting of lower and upper bounds on the integer variables that can be interpreted as branch-and-bound nodes for (1.1) as well as (MIP(T,S)). We define the following problems associated with N^k:

$(MISOC^k)$	mixed integer SOCP with bounds of N^k
(SOC^k)	continuous relaxation of $(MISOC^k)$
$(MIP^k(T,S))$	MIP outer approximation of $(MISOC^k)$
$(LP^k(T,S))$	continuous relaxation of $(MIP^k(T,S))$

Thus, if $(LP^k(T,S))$ is infeasible, (SOC^k), $(MISOC^k)$ and $(MIP^k(T,S))$ are also infeasible. The optimal objective function value of $(LP^k(T,S))$ is less or equal than the optimal objective function values of $(MIP^k(T,S))$ and (SOC^k) respectively, and these are less or equal than the optimal objective function value of $(MISOC^k)$. Thus, the algorithm stops searching the subtree of N^k either if the problem itself or its outer approximation becomes infeasible or if the objective function value of $(MIP^k(T,S))$ exceeds the optimal function value of a known feasible solution of (1.1). The latter case is expressed in the condition $c^Tx < c^T\bar{x},\ \forall \bar{x} \in T$ in $(MIP^k(T,S))$. The following hybrid algorithm integrates branch-and-bound and the outer approximation approach as proposed by Bonami et al. in [9] for convex differentiable MINLPs.

Algorithm 1 HYBRID OA/B-A-B FOR (1.1)

Input: Problem (1.1)
Output: Optimal solution x^* or indication of infeasibility.

Initialization: $CUB := \infty$, solve (SOC^0) with solution x^0,
if $((SOC^0)$ infeasible) STOP, problem infeasible
else set $S = \emptyset$, $T = \{x^0\}$ and solve (MIP(T,S))
endif

1. **if** ((MIP(T,S))infeasible) STOP, problem infeasible
else solution $x^{(1)}$ found:
if $(NLP(x_J^{(1)})$ feasible)
compute solution $\bar{x}$ of $(NLP(x_J^{(1)}))$, $T := T \cup \{\bar{x}\}$,
if $(c^T\bar{x} < CUB)$ $CUB = c^T\bar{x}$, $x^* = \bar{x}$ **endif**.
else compute solution $\bar{x}$ of $F(x_J^{(1)})$, $S := S \cup \{\bar{x}\}$.
endif
endif
$Nodes := \{N^0 = (lb^0 = l, ub^0 = u)\}$, $ll := 0$, $L := 10$, $i := 0$
2. **while** $Nodes \neq \emptyset$ **do** select N^k from $Nodes$, $Nodes := Nodes \setminus N^k$
2a. **if** $(ll = 0 \mod L)$ solve (SOC^k)
if $((SOC^k)$ feasible): solution $\bar{x}$, $T := T \cup \{\bar{x}\}$
if $(\bar{x}_J$ integer):
if $(c^T\bar{x} < CUB)$ $CUB = c^T\bar{x}$, $x^* = \bar{x}$ **endif**
go to 2.
endif
else go to 2.
endif
endif
2b. solve $(LP^k(T,S))$ with solution x^k
while $((LP^k(T,S))$ feasible) & $(x_J^k$ integer) & $(c^Tx^k < CUB)$
if $(NLP(x_J^k)$is feasible with solution $\bar{x})$ $T := T \cup \{\bar{x}\}$
if $(c^T\bar{x} < CUB)$ $CUB = c^T\bar{x}$, $x^* = \bar{x}$ **endif**
else solve $F(x_J^k)$ with solution $\bar{x}$, $S := S \cup \{\bar{x}\}$
endif
compute solution x^k of updated $(LP^k(T,S))$
endwhile
2c. **if** $(c^Tx^k < CUB)$ branch on variable $x_j^k \notin \mathbb{Z}$,

create $N^{i+1} = N^k$, with $ub_j^{i+1} = \lfloor x_j^k \rfloor$,
create $N^{i+2} = N^k$, with $lb_j^{i+2} = \lceil x_j^k \rceil$,
set $i = i + 2$, $ll = ll + 1$.
endif
endwhile

Note that if $L = 1$, x^k is set to $\bar{x}$ and Step 2b is omitted, Step 2 performs a nonlinear branch-and-bound search. If $L = \infty$ Algorithm 1 resembles a branch-and-bound based outer approximation algorithm. Convergence of the outer approximation approach in case of continuously differentiable constraint functions was shown in [7], Theorem 2. We now state convergence of Algorithm 1 for subdifferentiable SOCP constraints.
For this purpose, we first proof that the last integer assignment x_J^k is infeasible in the outer approximation conditions induced by the solution of a feasible subproblem $(NLP(x_J^k))$.

LEMMA 5.1. *Assume A1 and A2 hold. If $(NLP(x_J^k))$ is feasible with optimal solution $\bar{x}$ and dual solution $(\bar{s}, \bar{y})$. Then every x with $x_J = x_J^k$ satisfying the constraints $Ax = b$ and*

$$\begin{array}{rcl} -\|\bar{x}_{i1}\|x_{i0} + \bar{x}_{i1}^T x_{i1} & \leq & 0,\ i \in I_a(\bar{x}), \\ -x_{i0} & \leq & 0,\ i \in I_0(\bar{x}),\ \bar{s}_{i0} = 0, \\ -x_{i0} - \frac{1}{\bar{s}_{i0}}\bar{s}_{i1}^T x_{i1} & \leq & 0,\ i \in I_0(\bar{x}),\ \bar{s}_{i0} > 0,\ \bar{x} \in T, \end{array} \tag{5.1}$$

where I_a and I_0 are defined by (3.1), satisfies $c^T x \geq c^T \bar{x}$.

Proof. Assume x, with $x_J = \bar{x}_J$ satisfies $Ax = b$ and (5.1), namely

$$(\nabla g_i(\bar{x}))_{\bar{J}}^T (x_{\bar{J}} - \bar{x}_{\bar{J}}) \leq 0,\ i \in I_a(\bar{x}), \tag{5.2}$$

$$(\bar{\xi}_i)_{\bar{J}}^T (x_{\bar{J}} - \bar{x}_{\bar{J}}) \leq 0,\ i \in I_0(\bar{x}), \tag{5.3}$$

$$A(x - \bar{x}) = 0, \tag{5.4}$$

with $\bar{\xi}_i$ from Lemma 3.1 and where the last equation follows from $A\bar{x} = b$. Due to A2 we know that there exist $\mu \in \mathbb{R}^m$ and $\lambda \in \mathbb{R}_+^{|I_0 \cup I_a|}$ satisfying the KKT conditions (3.2) of $(NLP(x_J^k))$ in $\bar{x}$, that is

$$\begin{array}{rcll} -c_i & = & A_i^T \mu + \lambda_i \bar{\xi}_i, & i \in I_0(\bar{x}), \\ -c_i & = & A_i^T \mu + \lambda_i \nabla g_i(\bar{x}), & i \in I_a(\bar{x}), \\ -c_i & = & A_i^T \mu, & i \notin I_0(\bar{x}) \cup I_a(\bar{x}) \end{array} \tag{5.5}$$

with the subgradients $\bar{\xi}_i$ chosen from Lemma 3.1. Farkas' Lemma (cf. [21]) states that (5.5) is equivalent to the fact that as long as $(x - \bar{x})$ satisfies (5.2) - (5.4), then $c_{\bar{J}}^T (x_{\bar{J}} - \bar{x}_{\bar{J}}) \geq 0 \Leftrightarrow c_{\bar{J}}^T x_{\bar{J}} \geq c_{\bar{J}}^T \bar{x}_{\bar{J}}$ must hold. □
In the case that $(NLP(x_J^k))$ is infeasible, we can show that the subgradients (4.6) of Lemma 4.1 together with the gradients of the differentiable functions g_i in the solution of $(F(x_J^k))$ provide inequalities that separate the last integer solution.

LEMMA 5.2. *Assume A1 and A2 hold. If $(NLP(x_J^k))$ is infeasible and thus $(\bar{x}, \bar{u})$ solves $(F(x_J^k))$ with positive optimal value $\bar{u} > 0$, then every x satisfying the linear equalities $Ax = b$ with $x_J = x_J^k$, is infeasible in the constraints*

$$\begin{array}{rcl} -x_{i0} + \frac{\bar{x}_{i1}^T}{\|\bar{x}_{i1}\|} x_{i1} & \leq & 0,\ i \in I_{F1}(\bar{x}), \\ -x_{i0} - \frac{\bar{s}_{i1}^T}{\bar{s}_{i0}} x_{i1} & \leq & 0,\ i \in I_{F0}, \bar{s}_{i0} \neq 0, \\ -x_{i0} & \leq & 0,\ i \in I_{F0}, \bar{s}_{i0} = 0, \end{array} \tag{5.6}$$

where I_{F1} and I_{F0} are defined by (4.1) and $(\bar{s}, \bar{y})$ is the solution of the dual program $(F(x_J^k)$-D) of $(F(x_J^k))$.

Proof. The proof is done in analogy to Lemma 1 in [7]. Due to assumption A1 and A2, the optimal solution of $(F(x_J^k))$ is attained. We further know from Lemma 4.1, that there exist $\lambda_{g_i} \geq 0$, with $\sum_{i \in I_F} \lambda_{g_i} = 1$, μ_A and μ_J satisfying the KKT conditions

$$\sum_{i \in I_{F1}} \nabla g_i(\bar{x}) \lambda_{g_i} + \sum_{i \in I_{F0}} \xi_i^n \lambda_{g_i} + A^T \mu_A + I_J^T \mu_J = 0 \tag{5.7}$$

in $\bar{x}$ with subgradients (4.6). To show the result of the lemma, we assume now that x, with $x_J = x_J^k$, satisfies $Ax = b$ and conditions (5.6) which are equivalent to

$$\begin{array}{r} g_i(\bar{x}) + \nabla g_i(\bar{x})^T (x - \bar{x}) \leq 0,\ i \in I_{F1}(\bar{x}), \\ g_i(\bar{x}) + \xi_i^{n,T} (x - \bar{x}) \leq 0,\ i \in I_{F0}(\bar{x}). \end{array}$$

We multiply the inequalities by $(\lambda_g)_i \geq 0$ and add all inequalities. Since $g_i(\bar{x}) = \bar{u}$ for $i \in I_F$ and $\sum_{i \in I_F} \lambda_{g_i} = 1$ we get

$$\sum_{i \in I_{F1}} (\lambda_{g_i} \bar{u} + \lambda_{g_i} \nabla g_i(\bar{x})^T (x - \bar{x})) + \sum_{i \in I_{F0}} (\lambda_{g_i} \bar{u} + \lambda_{g_i} \xi_i^{n,T} (x - \bar{x})) \leq 0$$

$$\Leftrightarrow \qquad \bar{u} + \left(\sum_{i \in I_{F1}} \lambda_{g_i} \nabla g_i(\bar{x}) + \sum_{i \in I_{F0}} (\lambda_{g_i} \xi_i^n) \right)^T (x - \bar{x}) \leq 0.$$

Insertion of (5.7) yields

$$\begin{array}{rcl} & \bar{u} + (-A^T \mu_A - I_J^T \mu_J)^T (x - \bar{x}) & \leq 0 \\ \Leftrightarrow^{Ax = A\bar{x} = b} & \bar{u} - \mu_J^T (x_J - \bar{x}_J) & \leq 0 \\ \Leftrightarrow^{x_J = x_J^k = \bar{x}_J} & \bar{u} & \leq 0. \end{array}$$

This is a contradiction to the assumption $\bar{u} > 0$. ☐

Thus, the solution $\bar{x}$ of $(F(x_J^k))$ produces new constraints (5.6) that strengthen the outer approximation such that the integer solution x_J^k is no longer feasible. If $(NLP(x_J^k))$ is infeasible, the active set $I_F(\bar{x})$ is not empty and thus, at least one constraint (5.6) can be added.

THEOREM 5.1. *Assume A1 and A2. Then Algorithm 1 terminates in a finite number of steps at an optimal solution of (1.1) or with the indication, that it is infeasible.*

Proof. We show that no integer assignment x_J^k is generated twice by showing that $x_J = x_J^k$ is infeasible in the linearized constraints created in the solutions of $(NLP(x_J^k))$ or $(F(x_J^k))$. The finiteness follows then from the boundedness of the feasible set. A1 and A2 guarantee the solvability, validity of KKT conditions and primal-dual optimality of the nonlinear subproblems $(NLP(x_J^k))$ and $(F(x_J^k))$. In the case, when $(NLP(x_J^k))$ is feasible with solution $\bar{x}$, Lemma 5.1 states that every $\tilde{x}$ with $\tilde{x}_J = \hat{x}_J$ must satisfy $c^T\tilde{x} \geq c^T\bar{x}$ and is thus infeasible in the constraint $c^T\tilde{x} < c^T\bar{x}$ included in $(LP^k(T,S))$. In the case, when $(NLP(x_J^k))$ is infeasible, Lemma 5.2 yields the result for $(F(x_J^k))$. □

Modified Algorithm avoiding A2. We now present an adaption of Algorithm 1 which is still convergent if the convergence assumption A2 is not valid for every subproblem. Assume N^k is a node such that A2 is violated by $(NLP(x_J^k))$ and assume x with integer assignment $x_J = x_J^k$ is feasible for the updated outer approximation. Then the inner while-loop in step 2b becomes infinite and Algorithm 1 does not converge. In that case we solve the SOCP relaxation (SOC^k) in node N^k. If that problem is not infeasible and has no integer feasible solution, we branch on the solution of this SOCP relaxation to explore the subtree of N^k. Hence, we substitute step 2b by the following step.

2b'. solve $(LP^k(T,S))$ with solution x^k, set `repeat = true`.
    while $(((LP^k(T,S))$ feasible) & (x_J^k integer) & ($c^Tx^k < CUB$) & repeat)
        save $x_J^{old} = x_J^k$
        if $(NLP(x_J^k)$is feasible with solution $\bar{x}$)
            $T := T \cup \{\bar{x}\}$,
            if ($c^T\bar{x} < CUB$) $CUB = c^T\bar{x}$, $x^* = \bar{x}$ **endif**
        else compute solution $\bar{x}$ of $F(x_J^k)$, $S := S \cup \{\bar{x}\}$
        endif
        compute solution x^k of updated $(LP^k(T,S))$
        if ($x_J^{old} == x_J^k$) set repeat = false **endif**
    endwhile
    if(!repeat)
        solve nonlinear relaxation (SOC^k) at the node N^k with solution $\bar{x}$
        $T := T \cup \{\bar{x}\}$
        if ($\bar{x}_J$ integer): **if** $c^T\bar{x} < CUB$: $CUB = c^T\bar{x}$, $x^* = \bar{x}$ **endif**
        go to 2.
        else set $x^k = \bar{x}$.
        endif

endif

Note that every subgradient of a conic constraint provides a valid linear outer approximation of the form (2.1). Thus, in the case that we cannot identify the subgradients satisfying the KKT system of $(NLP(x_J^k))$, we take an arbitrary subgradient to update the linear outer approximation $(LP^k(T,S))$.

LEMMA 5.3. *Assume A1 holds. Then Algorithm 2, which is Algorithm 1, where Step 2b is replaced by 2b', terminates in a finite number of steps at an optimal solution of (1.1) or with the indication that it is infeasible.*

Proof. If A2 is not satisfied, we have no guarantee that the linearization in the solution $\bar{x}$ of $(NLP(x_J^k))$ separates the current integer solution. Hence, assume the solution of $(LP^k(T,S))$ is integer feasible with solution x^k and the same integer assignment x_J^k is optimal for the updated outer approximation $(LP^k(T \cup \{\bar{x}\}, S))$ or $(LP^k(T, S \cup \{\bar{x}\}))$. Then the nonlinear relaxation (SOC^k) is solved at the current node N^k. If the problem (SOC^k) is not infeasible and its solution is not integer, the algorithm branches on its solution producing new nodes N^{i+1} and N^{i+2}. These nodes are again searched using Algorithm 1 as long as the situation of repeated integer solutions does not occur. Otherwise it is again branched on the solution of the continuous relaxation. If this is done for a whole subtree, the algorithm coincides with a nonlinear branch-and-bound search for this subtree which is finite due to the boundedness of the integer variables. □

Remarks. The convergence result of Theorem 5.1 can be directly extended to any outer approximation approach for (1.1) which is based on the properties of (MIP(T,S))(and thus $(LP^k(T,S))$) proved in Lemma 5.1 and Lemma 5.2. In particular, convergence of the classical outer approximation approach as well as the Generalized Benders Decomposition approach (cf. [6], [4] or [28]) is naturally implied.
Furthermore, our convergence result can be generalized to arbitrary mixed integer programming problems with subdifferentiable convex constraint functions, if it is possible to identify the subgradients that satisfy the KKT system in the solutions of the associated nonlinear subproblems.

6. Numerical experiments. We implemented the modified version of the outer approximation approach Algorithm 2 ('B&B-OA') as well as a nonlinear branch-and-bound approach ('B&B'). The SOCP problems are solved with our own implementation of an infeasible primal-dual interior point approach (cf. [27], Chapter 1), the linear programs are solved with CPLEX 10.0.1.
We report results for mixed 0-1 formulations of different ESTP test problems (instances t_4*, t_5*) from Beasley's website [22] (cf. [17]) and some problems arising in the context of turbine balancing (instances Test*). The data sets are available on the web [30]. Each instance was solved using the

Problem	n	m	noc	$\|J\|$	m_oa (L=10)	m_oa (L=10000)
t4_nr22	67	50	49	9	122	122
t4_nrA	67	50	49	9	213	231
t4_nrB	67	50	49	9	222	240
t4_nrC	67	50	49	9	281	272
t5_nr1	132	97	96	18	1620	1032
t5_nr21	132	97	96	18	2273	1677
t5_nrA	132	97	96	18	1698	998
t5_nrB	132	97	96	18	1717	1243
t5_nrC	132	97	96	18	1471	1104
Test07	84	64	26	11	243	243
Test07_an	84	63	33	11	170	170
Test54	366	346	120	11	785	785
Test07GF	87	75	37	12	126	110
Test54GF	369	357	131	12	1362	1362
Test07_lowb	212	145	153	56	7005	2331
Test07_lowb_an	210	145	160	56	1730	308

Table 1
Problem sizes ($m, n, noc, |J|$) and maximal constraints of LP approximation (m_oa)

nonlinear branch-and-bound algorithm as well as Algorithm 2, once for the choice $L = 10$ and for the choice $L = 10000$ respectively.
We used best first node selection and pseudocost branching in the nonlinear branch-and-bound approach and depth first search as well as most fractional branching in Algorithm 2, since those performed best in former tests.
Table 1 gives an overview of the problem dimensions according to the notation in this paper. We also list for each problem the number of constraints of the largest LP relaxation solved during Algorithm 2 ('m_oa'). As depicted in Algorithm 2, every time a nonlinear subproblem is solved, the number of constraints of the outer approximation problem grows by the number of conic constraints active in the solution of that nonlinear subproblem. Thus, the largest fraction of linear programs solved during the algorithm have significantly fewer constraints than ('m_oa').
For each algorithm Table 2 displays the number of solved SOCP nodes and LP nodes whereas Table 3 displays the run times. A comparison of the branch-and-bound approach and Algorithm 2 on the basis of Table 2 shows, that the latter algorithm solves remarkable fewer SOCP problems. Table 3 displays that for almost all test instances, the branch-and-bound based outer approximation approach is preferable regarding running times, since the LP problems stay moderately in size.

Problem	B&B (SOCP)	B&B-OA (L=10) (SOCP/LP)	B&B-OA (L=10000) (SOCP/LP)
t4_nr22	31	9/15	**9/15**
t4_nrA	31	**19/39**	20/40
t4_nrB	31	**20/40**	21/41
t4_nrC	31	26/**43**	**25/43**
t5_nr1	465	120/745	**52/720**
t5_nr21	613	170/**957**	**88**/1010
t5_nrA	565	140/**941**	**50**/995
t5_nrB	395	105 /**519**	**64**/552
t5_nrC	625	115/761	**56/ 755**
Test07	13	**8/20**	**8/20**
Test07_an	7	**5/9**	**5/9**
Test54	7	**5/9**	**5/9**
Test07GF	41	5/39	**2/35**
Test54GF	37	11/68	**9/63**
Test07_lowb	383	392/3065	**115/2599**
Test07_lowb_an	1127	128/**1505**	**9**/1572

TABLE 2
Number of solved SOCP/LP problems

For $L = 10$, at every 10th node an additional SOCP problem is solved whereas for $L = 10000$, for our test set no additional SOCP relaxations are solved. In comparison with $L = 10000$, for $L = 10$ more (11 out of 16 instances) or equally many (3 out of 16 instances) SOCP problems are solved, whereas the number of solved LP problems is decreased only for 6 out of 16 instances. Moreover, the number of LPs spared by the additional SOCP solves for $L = 10$ is not significant in comparison with $L = 10000$ (compare Table 2) and the sizes of the LPs for $L = 10000$ stay smaller in most cases, since fewer linearizations are added (compare Table 1). Hence, with regard to running times, the version with $L = 10000$ outperforms $L = 10$ for almost all test instances, compare Table 3. Thus, for the problems considered, Algorithm 2 with $L = 10000$, i.e., without solving additional SOCPs, achieves the best performance in comparison to nonlinear branch-and-bound as well as Algorithm 1 with $L = 10$.

In addition to the above considered instances, we tested some of the classical portfolio optimization instances provided by Vielma et al. [29] using Algorithm 2 with $L = 10000$. For each problem, we report in Table 4 the dimension of the MISOCP formulation, the dimension of the largest relaxation solved by our algorithm and the dimension of the a priori LP

Problem	B&B	B&B-OA (L=10)	B&B-OA (L=10000)
t4_nr22	2.83	**0.57**	0.63
t4_nrA	2.86	**1.33**	1.44
t4_nrB	2.46	1.72	**1.71**
t4_nrC	2.97	1.54	**2.03**
t5_nr1	128.96	42.22	**29.58**
t5_nr21	139.86	61.88	**20.07**
t5_nrA	128.37	53.44	**17.04**
t5_nrB	77.03	36.55	**18.94**
t5_nrC	150.79	44.57	**16.11**
Test07	0.42	0.28	**0.26**
Test07_an	**0.11**	0.16	0.14
Test54	4.4	1.33	**1.40**
Test07GF	2.26	0.41	**0.24**
Test54GF	32.37	6.95	**5.53**
Test07_lowb	244.52	499.47	**134.13**
Test07_lowb_an	893.7	128.44	**14.79**

TABLE 3
Run times in seconds

relaxation with accuracy 0.01 that was presented in [11]. For a better comparison, we report the number of columns plus the number of linear constraints as it is done in [11]. The dimensions of the largest LP relaxations solved by our approach are significantly smaller than the dimensions of the LP approximations solved by [11]. Furthermore, in the lifted linear programming approach in [11], every LP relaxation solved during the algorithm is of the specified dimension. In our approach most of the solved LPs are much smaller than the reported maximal dimension. In Table 5 we report the run times and number of solved nodes problems for our algorithm (Alg.2). For the sake of completeness we added the average and maximal run times reported in [11] although it is not an appropriate comparison since the algorithms have not been tested on similar machines. Since our implementation of an interior SOCP solver is not as efficient as a commercial solver like CPLEX which is used in [11], a comparison of times is also difficult. But the authors of [11] report that solving their LP relaxations usually takes longer than solving the associated SOCP relaxation. Thus, we can assume that due to the low dimensions of the LPs solved in our approach and the moderate number of SOCPs, our approach is likely to be faster when using a more efficient SOCP solver.

7. Summary. We presented a branch-and-bound based outer approximation approach using subgradient based linearizations. We proved con-

Problem	n + m	$\|J\|$	n+ m_oa [Alg.2]	nums+cols [11]
classical_20_0	105	20	137	769
classical_20_3	105	20	129	769
classical_20_4	105	20	184	769
classical_30_0	155	30	306	1169
classical_30_1	155	30	216	1169
classical_30_3	155	30	207	1169
classical_30_4	155	30	155	1169
classical_40_0	205	40	298	1574
classical_40_1	205	40	539	1574
classical_40_3	205	40	418	1574
classical_50_2	255	50	803	1979
classical_50_3	255	50	867	1979

TABLE 4
Dimension (m+n) and maximal LP approximation ($m\_oa + n$) (portfolio instances)

Problem	Sec. [Alg.2]	Nodes [Alg.2] (SOCP/LP)	Sec. [11] (average)	Sec. [11] (max)
classical_20_0	2.62	10/75	0.29	1.06
classical_20_3	0.62	2/9	0.29	1.06
classical_20_4	5.70	32/229	0.29	1.06
classical_30_0	52.70	119/2834	1.65	27.00
classical_30_1	16.13	30/688	1.65	27.00
classical_30_3	8.61	20/247	1.65	27.00
classical_30_4	0.41	1/0	1.65	27.00
classical_40_0	46.07	51/1631	14.84	554.52
classical_40_1	361.62	292/15451	14.84	554.52
classical_40_3	138.28	171/5222	14.84	554.52
classical_50_2	779.74	496/ 19285	102.88	1950.81
classical_50_3	1279.61	561/36784	102.88	1950.81

TABLE 5
Run times and node problems (portfolio instances)

vergence under a constraint qualification that guarantees strong duality of the occurring subproblems and extended the algorithm such that this assumption can be relaxed. We presented numerical experiments for some application problems investigating the performance of the approach in terms of solved LP, SOCP problems and running times. We also investigated the sizes of the linear approximation problems. Comparison to a nonlinear branch-and-bound algorithm showed that the outer approximation approach solves almost all problems in significantly shorter running times and that its performance is best when not solving additional SOCP relaxations. In comparison to the outer approximation based approach by Vielma et al. in [11], we observed that the dimensions of our LP relaxations are significantly smaller which makes our approach competitive.

Acknowledgements. We would like to thank the referees for their constructive comments that were very helpful to improve this paper.

REFERENCES

[1] R. Fletcher and S. Leyffer, *Solving Mixed Integer Nonlinear Programs by Outer Approximation*, in Mathematical Programming, 1994, 66: pp. 327-349.
[2] R. A. Stubbs and S. Mehrotra, *A branch-and-cut method for 0-1 mixed convex programming* in Mathematical Programming, 1999, 86: pp.515-532
[3] I. Quesada and I.E. Grosmann, *An LP/NLP based Branch and Bound Algorithm for Convex MINLP Optimization Problems*, in Computers and Chemical Engineering, 1992, 16:(10,11) pp. 937-947 13
[4] A. M. Geoffrion, *Generalized Benders Decomposition*, in Journal of Optimization Theory and Applications, 1972, 10(4):pp.237-260
[5] M.T. Çezik and G. Iyengar, *Cuts for Mixed 0-1 Conic Programming*, in Mathematical Programming, Ser. A, 2005, 104: pp. 179-200
[6] M. A. Duran and I. E. Grossmann, *An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs*, in Mathematical Programming, 1986,36: pp. 307-339
[7] R. Fletcher and S. Leyffer, *Solving Mixed Integer Nonlinear Programs by Outer Approximation*, in Mathematical Programming, 1994, 66: pp. 327-349.
[8] F. Alizadeh and D. Goldfarb, *Second-Order Cone Programming*, RUTCOR, Rutgers Center for Operations Research,Rutgers University, New Jersey, 2001
[9] P. Bonami and L.T. Biegler and A.R.Conn and G. Cornuejols and I.E. Grossmann and C.D.Laird and J. Lee and A.Lodi and F.Margot and N.Sawaya and A. Wächter , *An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs*, IBM Research Division, New York, 2005
[10] R. A. Stubbs and S. Mehrotra, *Generating Convex Polynomial Inequalities for Mixed 0-1 Programs*, Journal of global optimization, 2002, 24: pp. 311-332
[11] J. P. Vielma and S. Ahmed and G. L. Nemhauser, *A Lifted Linear Programming Branch-and-Bound Algorithm for Mixed Integer Conic Quadratic Programs*, INFORMS Journal on Computing, 2008,20(3): pp. 438-450
[12] A. Atamtürk and V. Narayanan, *Lifting for Conic Mixed-Integer Programming*, BCOL Research report 07.04, 2007
[13] A. Atamtürk and V. Narayanan, *Cuts for Conic Mixed-Integer Programming*, Mathematical Programming, Ser. A, DOI 10.1007/s10107-008-0239-4, 2007
[14] A. Ben-Tal and A. Nemirovski, *On Polyhedral Approximations of the Second-Order Cone*, in Mathematics of Operations Research, 2001, 26(2):pp. 193–205
[15] E. Balas and S. Ceria and G. Cornuéjols, *A lift-and-project cutting plane algorithm for mixed 0-1 programs*, in Mathematical Programming, 1993, 58: pp. 295-324
[16] M. Fampa and N. Maculan, *A new relaxation in conic form for the Euclidean Steiner Tree Problem in* $\mathbb{R}^n$, in RAIRO Operations Research, 2001,35: pp. 383-394
[17] J. Soukup and W.F. Chow, *Set of test problems for the minimum length connection networks*, in ACM SIGMAP Bulletin, 1973, 15: pp. 48-51

[18] D. Bertsimas and R. Shioda, *Algorithm for cardinality-constrained quadratic optimization*, in Computational Optimization and Applications, 2007,91: pp. 239-269
[19] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Studies in Applied Mathematics, 2001
[20] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970
[21] C. Geiger and C. Kanzow, *Theorie und Numerik restringierter Optimierungsaufgaben*, Springer Verlag Berlin Heidelberg New York, 2002
[22] J. E. Beasley, *OR Library: Collection of test data for Euclidean Steiner Tree Problems*,
`http://people.brunel.ac.uk/ mastjjb/jeb/orlib/esteininfo.html`,
[23] P. Belotti and P. Bonami and J. J. Forrest and L. Ladanyi and C. Laird and J. Lee and F. Margot and A. Wächter, *BonMin*, `http://www.coin-or.org/Bonmin/`
[24] R. Fletcher and S. Leyffer, *User Manual of filterSQP*,
`http://www.mcs.anl.gov/ leyffer/papers/SQP_manual.pdf`
[25] C. Laird and A. Wächter, *IPOPT*,
`https://projects.coin-or.org/Ipopt`
[26] K. Abhishek and S. Leyffer and J. T. Linderoth, *FilMINT: An Outer Approximation-Based Solver for Nonlinear Mixed Integer Programs*, Argonne National Laboratory, Mathematics and Computer Science Division,2008
[27] S. Drewes, *Mixed Integer Second Order Cone Programming*, PhD Thesis, submitted April, 2009
[28] Bonami, P. and Kilinc, M. and Linderoth, J., *Algorithms and Software for Convex Mixed Integer Nonlinear Programs* 2009
[29] J.P. Vielma *Portfolio Optimization Instances*
`http://www2.isye.gatech.edu/ jvielma/portfolio/`
[30] S. Drewes *MISOCP Test Instances*
`https://www3.mathematik.tu-darmstadt.de/index.php?id=491`