Industrial and
Systems Engineering

The first heuristic specifically for mixed-integer second-order cone optimization

Sertalp B. Çay

Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

Imre Pólik

Operations Research R & D, SAS Institute, Cary, NC, USA

Tamás Terlaky

Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

ISE Technical Report 18T-002

Lehigh University

The first heuristic specifically for mixed-integer second-order cone optimization

Sertalp B. Çay, Imre Pólik, Tamás Terlaky

January 22, 2018

Abstract

Mixed-integer second-order cone optimization (MISOCO) has become very popular in the last decade. Various aspects of solving these problems in Branch and Conic Cut (BCC) algorithms have been studied in the literature. This study aims to fill a gap and provide a novel way to find feasible solutions early in the BCC algorithm. Such solutions have a huge impact on reducing tree size and proving the feasibility of the problem. Despite the rich literature on mixed-integer linear optimization (MILO), there is no work on heuristic methods specific to MISOCO problems. In this work, we use the optimal Jordan frames of a second-order cone optimization (SOCO) subproblem to generate MILO rounding problems. These rounding problems can provide feasible solutions for the original MISOCO instances in a relatively short time. Based on extensive computational experiments, the proposed heuristics provide feasible solutions for 1327 out of 1328 problems in CBLIB and QPLIB test problems at the root node.

Keywords— mixed-integer conic optimization, branch and bound, heuristic methods

1 Introduction

Primal heuristics are one of the most important elements of search tree methods. Despite the fact that primal heuristics are not exact methods, their contribution to the efficiency of commercial solvers is significant. Therefore, commercial MILO solvers are packed with heuristics [5, 24, 29, 30].

MISOCO formulations consist of a linear objective and a set of linear and conic constraints. A MISOCO formulation in the standard form can be written as follows:

$$\begin{array}{llll} \min & c^\top x & & \\ \text{s.t.} & Ax & = & b, \\ & x & \in & \mathcal{K}, \\ & x_j & \in & \mathbb{Z} \qquad \forall j \in J \subseteq N, \end{array} \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$ is a full row rank matrix, $c \in \mathbb{R}^n$, and $b \in \mathbb{R}^m$, $N = \{1, 2, \ldots, n\}$, $\mathcal{K}$ is the Cartesian product of second-order cones of various dimensions, i.e., $\mathcal{K} = \mathbb{L}^{n_1} \times \mathbb{L}^{n_2} \times \cdots \times \mathbb{L}^{n_k}$, where $\mathbb{L}^{n_i} = \{x^i \in \mathbb{R}^{n_i} \mid x^i_1 \geq \|x^i_{2:n_i}\|\}$, for $i = 1, \ldots, k$, with $\sum_{i=1}^k n_i = n$ for $x = \left((x^1)^\top, (x^2)^\top, \ldots, (x^k)^\top\right)^\top$, and $x^i \in \mathbb{R}^{n_i}$. The development of solution methodologies for MISOCO problems is an active research area. As MISOCO is a generalization of MILO, researchers mainly focus on translating existing MILO techniques to MISOCO. The main reason behind the popularity of MISOCO problems is twofold. The first reason is the availability of MISOCO formulations for a variety of problems from different sectors, such as portfolio optimization problems [12, 19] and option pricing problems [33] from finance, the turbine balancing problem [22] from energy, and the stereotactic surgery treatment planning with isocenter selection problem [28] from healthcare. Another reason is the existence of efficient solution methodologies to solve underlying SOCO subproblems.

The ongoing research has been focused on generating valid inequalities for MISOCO problems [6, 7, 8, 9, 10, 11, 21, 31, 32, 36]. Despite recent advances in warm-start of MISOCO problems [20], vital elements of a full BCC framework are still missing, such as pre-processing and primal heuristics. Notably MILO has a remarkably rich literature on this topic. Since there is no available study on heuristics specific to MISOCO, we give a brief review of the MILO and MINLO literature.

Primal heuristics play an important role in state-of-the-art MILO solvers. Their main role is to provide an upper bound early in the search tree. This upper bound proves feasibility of the problem and can help reduce search tree size significantly by pruning nodes by bound early [3]. There are three types of primal heuristics. The first set of heuristics is called *diving heuristics*, which dive inside the search tree with the aim of finding a feasible MILO solution as quickly as possible. The second set of heuristics is called *rounding heuristics*, where the aim is to generate a feasible solution by rounding a fractional LO solution. The third set of heuristics is called *improvement heuristics*, where one or more primal feasible solutions are used to construct a better feasible solution. For an extensive discussion, see Achterberg [1] and Berthold [14].

Fischetti et al. [25] propose a heuristic called feasibility pump (FP). In simple terms, FP generates two sets of solutions: the first set consists of points that satisfy feasibility constraints except integrality, whereas the second set consists of points that are integer but possibly infeasible. These solutions are generated consecutively by using rounding and then solving an auxiliary problem. This heuristic is shown to be effective for binary MILO problems. FP's efficiency has led to several variations of the heuristic are discussed in the literature. Bertacco et al. [13] work on improving the efficiency of FP for general MILO problems. Moreover, they provide a restart method to prevent cycling. Achterberg and Berthold [2] propose a variation of FP that takes the objective function into consideration when finding solutions. Their aim is to find a better feasible solution for MILO instead of an arbitrary one, and their modification is shown to be more effective for the majority of the test problems in terms of providing a better bound. Bonami et al. [18] and Bonami and Gonçalves [17] extend FP for convex MINLO problems in different ways. Bonami and Gonçalves [17] discussed extensions of several primal heuristics for convex MINLO problems and showed that variants of FP can provide a solution for 93—94% of instances. Primal heuristics for MINLO problems help them to reduce the total solution time about 11% on average. Finally, Berthold [15] gives an extensive review of heuristics for MINLO and presents a variation of FP for non-convex MINLO problems.

Berthold et al. [16] extend the SCIP solver to solve mixed-integer quadratic optimization (MIQO) problems by using two simple primal heuristics. They compare SCIP against other solvers on MIQO problems, where some of them are MISOCO instances available in the CBLIB [26] test set.

It is apparent that many of the available MILO heuristics can be used for MISOCO problems. However, MISOCO formulations allow us to explore further heuristics that are specific to MISOCO. To our knowledge, there is no research on development of heuristics for general MISOCO problems.

The purpose of this paper is to present novel heuristics that are specific to MISOCO problems to generate feasible solutions in BCC search tree. We consider a general MISOCO formulation, where the Jordan frames of the SOCO subproblem can be obtained easily. These Jordan frames are used to create linear constraints that enforce conic feasibility in a limited feasible region. By solving the rounding MILO problems, it is possible to generate a feasible solution for many problems in practice. Then the generated solution can be improved by changing continuous variables. This way, we can generate a feasible solution even at the root node of a BCC search tree, which helps prune more nodes, prove feasibility, and provide an upper bound to estimate the optimality gap throughout the search. Our method consists of two parts: in the first part (rounding), we solve several SOCO problems to construct a feasible MISOCO solution and in the second part (improvement), we use this solution to improve the solution.

The rest of the paper is structured as follows: In Section 2 we give preliminaries. In Section 3, we give descriptions of conic rounding heuristics and their translation to quadratic optimization formulations. This is followed by the results of the numerical experiments on CBLIB test problems in Section 4. Finally, summary of the paper, implications, and future research directions are discussed in Section 5.

2 Preliminaries

2.1 Jordan frames

A quick review of related Jordan algebra concepts is given in this subsection, see [4, 23] for details. Given a second-order cone $\mathbb{L}^{n_i} \subseteq \mathbb{R}^{n_i}$, a vector x can be decomposed as follows:

$$x^i = \lambda_i^+ f_i^+ + \lambda_i^- f_i^-,$$

where

$$\lambda_i^+ = x_1^i + \|x_{2:n_i}^i\|, \qquad \lambda_i^- = x_1^i - \|x_{2:n_i}^i\|,$$
$$f_i^+ = \frac{1}{2}\begin{pmatrix} 1 \\ \frac{x_{2:n_i}^i}{\|x_{2:n_i}^i\|} \end{pmatrix}, \qquad f_i^- = \frac{1}{2}\begin{pmatrix} 1 \\ -\frac{x_{2:n_i}^i}{\|x_{2:n_i}^i\|} \end{pmatrix}.$$

Here λ_i and f_i are called Jordan values and Jordan frames of the vector x^i, respectively. Vectors f_i^+ and f_i^- are always on the boundary of $\mathbb{L}^{n_i}$ and they are orthogonal to each other. For notational convenience let F be the matrix defined by the Jordan frames of vectors $x^i, i \in \{1, \dots, k\}$ and let λ be a vector of size $2k$, such as

$$F = \Bigg[\underbrace{\begin{matrix} f_1^+ & 0 & \dots & 0 \\ 0 & f_2^+ & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & f_k^+ \end{matrix}}_{F^+} \quad \underbrace{\begin{matrix} f_1^- & 0 & \dots & 0 \\ 0 & f_2^- & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & f_k^- \end{matrix}}_{F^-}\Bigg] \in \mathbb{R}^{n \times 2k},$$

$$\lambda = \begin{bmatrix}\lambda_1^+, & \lambda_2^+, & \dots & \lambda_k^+, & \lambda_1^-, & \lambda_2^-, & \dots & \lambda_k^-\end{bmatrix}^\top \in \mathbb{R}^n.$$

Now, we can write $x = F\lambda$. For the special case of $\|x_{2:n_i}\| = 0$, we can choose any Jordan frame pair f_i^+, f_i^-, as the vector x^i can be represented when $\lambda_i^+ = \lambda_i^- = x_1^i$ in this case. Choosing a random Jordan frame for this special case has no impact on the heuristic.

Notice that for a given f_i^+ and f_i^-, the vector $x^i = \lambda_i^+ f_i^+ + \lambda_i^- f_i^-$ is in $\mathbb{L}^{n_i}$ if and only if $\lambda_i^+, \lambda_i^- \geq 0$. This is a key observation that ensures conic feasibility.

2.2 Rounding problems

Terlaky and Pólik [35] and Pólik and Góez [34] present rounding problems for SOCO problems. Rounding problems are originally introduced to provide a way to get a numerically better solution at the end of IPMs when solving SOCO instances. We give a brief review of rounding problems and their duals in this subsection. For the following derivations, consider the continuous relaxation of a MISOCO problem,

$$\begin{array}{rrcl} \min & c^\top x & & \\ \text{s.t.} & Ax & = & b, \\ & x & \in & \mathcal{K}, \end{array}$$

and its dual

$$\begin{array}{rrcl} \max & b^\top y & & \\ \text{s.t.} & A^\top y + z & = & c, \\ & z & \in & \mathcal{K}. \end{array}$$

Let F_P be the Jordan frame obtained from the primal variable x and F_D the Jordan frame obtained from the dual slack variable[1] z. The primal rounding problem replaces the primal variable x with $F_P\lambda$ and enforces conic feasibility with non-negativity of the Jordan values λ as follows:

$$\begin{array}{rrcl} \min & c^\top F_P \lambda & & \\ \text{s.t.} & AF_P\lambda & = & b, \\ & \lambda & \geq & 0. \end{array} \tag{PR}$$

[1] If x and z are an optimal primal-dual solution, then F_P and F_D can be chosen the same.

Symmetrically, the dual rounding problem replaces the dual slack variable z with $F_D\kappa$ and enforces conic feasibility with non-negativity of the Jordan values κ. It is written as follows:

$$\begin{array}{lrcl} \max & b^\top y & & \\ \text{s.t.} & A^\top y + F_D\kappa & = & c, \\ & \kappa & \geq & 0. \end{array} \tag{DR}$$

$$\begin{array}{lrcl} \min & (c^\top F_P)\lambda & & \\ \text{s.t.} & (AF_P)\lambda & = & b, \\ & \lambda & \geq & 0 \end{array} \tag{PR}$$
$[m] \times [2k]$

$\overset{\text{dual}}{\Leftrightarrow}$

$$\begin{array}{lrcl} \max & b^\top y & & \\ \text{s.t.} & F_P^\top A^\top y + u & = & F_P^\top c, \\ & u & \geq & 0 \end{array} \tag{D-PR}$$
$[2k] \times [m+2k]$

(for $x = F_P\lambda$) $\rotatebox{90}{$\supseteq$}$ $\qquad$ $\rotatebox{90}{$\subseteq$}$

$$\begin{array}{lrcl} \min & c^\top x & & \\ \text{s.t.} & Ax & = & b, \\ & x & \in & \mathcal{K} \end{array} \tag{P-SOCO}$$
$[m] \times [n]$

$\overset{\text{dual}}{\Leftrightarrow}$

$$\begin{array}{lrcl} \max & b^\top y & & \\ \text{s.t.} & A^\top y + z & = & c, \\ & z & \in & \mathcal{K} \end{array} \tag{D-SOCO}$$
$[n] \times [m+n]$

$\rotatebox{90}{$\supseteq$}$ $\qquad$ (for $z = F_D\kappa$) $\rotatebox{90}{$\subseteq$}$

$$\begin{array}{lrcl} \min & c^\top x & & \\ \text{s.t.} & Ax & = & b, \\ & F_D^\top x & \geq & 0 \end{array} \tag{D-DR}$$
$[m+2k] \times [n]$

$\overset{\text{dual}}{\Leftrightarrow}$

$$\begin{array}{lrcl} \max & b^\top y & & \\ \text{s.t.} & A^\top y + F_D\kappa & = & c, \\ & \kappa & \geq & 0 \end{array} \tag{DR}$$
$[n] \times [m+2k]$

Figure 1: Feasibility and duality relationship between SOCO and rounding LO problems

Notice that (PR) and (DR) are not duals of each other. We introduce their duals, the dual of the primal rounding (D-PR) and the dual of the dual rounding (D-DR) as given in Figure 1. Any solution to (PR) is always a feasible solution to the continuous relaxation of the MISOCO instance. Let F_P be the Jordan frame of a conic feasible solution $x \in \mathcal{K}$. Then, $\tilde{x} = F_P\lambda$ is a feasible solution to the continuous relaxation for any λ that is feasible to (PR). Similarly, we can find a feasible solution to original dual SOCO problem (D-SOCO) from any feasible solution to (DR). Let F_D be the Jordan frame of a conic feasible solution $z \in \mathcal{K}$ which is used to define (D-DR). Then, x is a feasible solution to (D-DR) for any x that is feasible to the continuous relaxation.

There is a weak duality relationship between (PR) and (DR). Let λ^* and $y^*_{(\text{DR})}$ be optimal solutions for (PR) and (DR), respectively. Then, $c^\top F_P\lambda^* \geq b^\top y^*_{(\text{DR})}$. This relationship can be exploited to prune nodes in a branch and bound tree, see Çay et al. [20]. It is important to emphasize that the strong duality property of the original SOCO relaxation is not assumed as it is not needed for any of our results.

3 Conic rounding heuristics

The main purpose of conic rounding heuristics is to provide a feasible solution for the original MISOCO problem. We begin this section with preliminaries on Jordan algebra and rounding problems. We use these components to describe four heuristics in the following subsections: the primal rounding heuristic, the dual rounding heuristic, the primal-dual rounding heuristic, and a hybrid heuristic.

3.1 The primal rounding heuristic

The main difficulties of finding a feasible solution for a MISOCO comes from two sources: integer variables and conic constraints. To deal with the latter source, one can use a restricted problem of the original instance by using a set of feasible Jordan frames. Primal rounding (PR) with integrality constraints can be used to generate feasible solutions to the original MISOCO instance. This idea is the underlying foundation of the *primal rounding heuristic*, where we solve a series of mixed-integer primal rounding and auxiliary SOCO problems to generate a feasible solution.

The primal rounding heuristic starts with an optimal solution of the continuous relaxation of MISOCO. The Jordan frame that corresponds to this solution is stored for the iterative steps. Then we solve the mixed-integer primal rounding problem (MIPR), which is written as follows:

$$\begin{array}{rrcll} \text{minimize:} & c^\top x & & & \\ \text{subject to:} & Ax & = & b, & \\ & x & = & F^*\lambda, & \\ & x_1^i & \geq & 0, & i \in 1,\dots,k \\ & x_j & \in & \mathbb{Z}, & j \in J \subseteq N \\ & \lambda & \in & \mathbb{R}^{2k}_+. & \end{array} \tag{MIPR}$$

If the (MIPR) problem is feasible, then its solution x^* is a feasible solution to the original MISOCO problem as well. This solution can be further improved by fixing the integer variables and solving the remaining SOCO problem to optimality. This problem is called fix-and-relax (FR) and is written as follows:

$$\begin{array}{rrcll} \text{minimize:} & c^\top x & & & \\ \text{subject to:} & Ax & = & b, & \\ & x_j & = & x_j^* & \forall j \in J \subseteq N, \\ & x & \in & \mathcal{K}. & \end{array} \tag{FR}$$

After solving (FR), we can update the best known upper bound. Moreover, the optimal solution $\bar{x}$ of (FR) is used to obtain a new Jordan frame. This way new solutions, which are obtained in consecutive iterations, are guaranteed to be bounded by this solution.

A penalty problem is solved to generate more Jordan frames for the problem in each iteration regardless of whether (MIPR) is feasible. Denote $\hat{F} = F^+ - F^-$ and the penalty problem is written as follows:

$$\begin{array}{rrcl} \text{minimize:} & \varphi \frac{c^\top}{\|c\|} x + (1-\varphi) \left\| \hat{F}^\top x \right\| & & \\ \text{subject to:} & Ax & = & b, \\ & x & \in & \mathcal{K}, \end{array}$$

where φ represents the tradeoff between the original objective and the penalty term. The reason for using the penalty term $\hat{F}^\top x$ is to obtain a different solution and a Jordan frame. The term $\|\hat{F}^\top x\|$ reaches its minimum value 0 if x is orthogonal to $\hat{F}$ and to its maximum value if the solution of the penalty problem x^p is equal to the (MIPR) solution. Because the transformation $x = F_P^\top \lambda$ and the constraint $\lambda \geq 0$ are used as a way to enforce conic feasibility and because new Jordan frames could be added to F, obtaining a larger feasible region is vital by obtaining different Jordan frames. Minimizing this term enlarges the feasible region for (MIPR) in consecutive iterations. The penalty term could be 1-norm, 2-norm, or infinity-norm and can be solved as a SOCO. The objective function of the penalty problem is similar to the objective proposed in [2], a penalty problem with a tradeoff between maximizing the feasible region and minimizing the original objective function. After the penalty problem is solved, (MIPR) is solved again with new Jordan frames, and the loop continues until a predefined termination criteria is reached, such as iteration or gap. All Jordan frames obtained in consecutive iterations are collected in a pool. Since the solution of the penalty problem is requested to be different from existing solutions, all Jordan frames can be added to the problem. The general form of the penalty problem (PEN) is written as follows:

$$\begin{array}{rrcl} \text{minimize:} & \varphi \frac{c^\top}{\|c\|} x + (1-\varphi) \sum_\ell \left\| \hat{F}_\ell^\top x \right\| & & \\ \text{subject to:} & Ax & = & b, \\ & x & \in & \mathcal{K}, \end{array} \tag{PEN}$$

where $\hat{F}_\ell$ is the penalty term that corresponds to the ℓ^{th} Jordan frame in the pool.

When (MIPR) is infeasible, it means that there are no feasible points in the restricted region defined by $x = F^* \lambda$. One can add more Jordan frames to F, which is equivalent to expanding the feasible region of (MIPR). In theory, one can add infinitely many unique Jordan frames to (MIPR) and solve the original MISOCO problem as a series of (MIPR) problems. Our aim is to keep this number at a reasonable level and still be able to produce feasible results. As shown in Section 4, we rarely need more than a few frames even for larger cones. In fact, three Jordan frames are enough to obtain a feasible solution for 98% of all test instances, so the generated Jordan frames are far from being an inner-approximation of SOCs.

An overview of the flow of the primal rounding heuristic is given in Figure 2. In the figure, C represents conic feasible solutions, I represents integer feasible solutions, and IC represents both integer and conic feasible solutions. Potential solution outcomes are indicated on the top right corner of blocks. The heuristic steps are given in Algorithm 1.

3.1.1 Example

We provide a numerical example in this subsection to describe how the primal rounding heuristic works in practice. Consider the following optimization model:

$$\begin{array}{rrcl} \text{minimize:} & 2x_1 + x_2 - 2x_3 & & \\ \text{subject to:} & 10\, x_1 + x_2 & = & 19, \\ & (x_1, x_2, x_3) & \in & \mathbb{L}^3, \\ & x_1, x_2 & \in & \mathbb{Z}. \end{array}$$

Following are the steps of the primal rounding heuristic, which are illustrated in Figure 3. Numerical values are given up to three digits of precision.

1. Solve SOCO subproblem:

$$\begin{array}{rrcl} \text{minimize:} & 2x_1 + x_2 - 2x_3 & & \\ \text{subject to:} & 10\, x_1 + x_2 & = & 19, \\ & (x_1, x_2, x_3) & \in & \mathbb{L}^3. \end{array}$$

   The solution of the SOCO subproblem is $x^s = (1.991, -0.907, 1.772)$. For illustration, see Figure 3a.

2. Obtain the Jordan frames. The F matrix is

$$F = \begin{bmatrix} 0.5 & 0.5 \\ -0.228 & 0.228 \\ 0.445 & -0.445 \end{bmatrix}.$$

3. Solve the (MIPR) problem:

$$\begin{array}{rrcl} \text{minimize:} & 2x_1 + x_2 - 2x_3 & & \\ \text{subject to:} & 10\, x_1 + x_2 & = & 19, \\ & x_1 & = & 0.5\lambda_1 + 0.5\lambda_2, \\ & x_2 & = & -0.228\lambda_1 + 0.228\lambda_2, \\ & x_3 & = & 0.445\lambda_1 - 0.445\lambda_2, \\ & x_1 & \geq & 0, \\ & x_1, x_2 & \in & \mathbb{Z}, \\ & \lambda_1, \lambda_2 & \in & \mathbb{R}_+. \end{array}$$

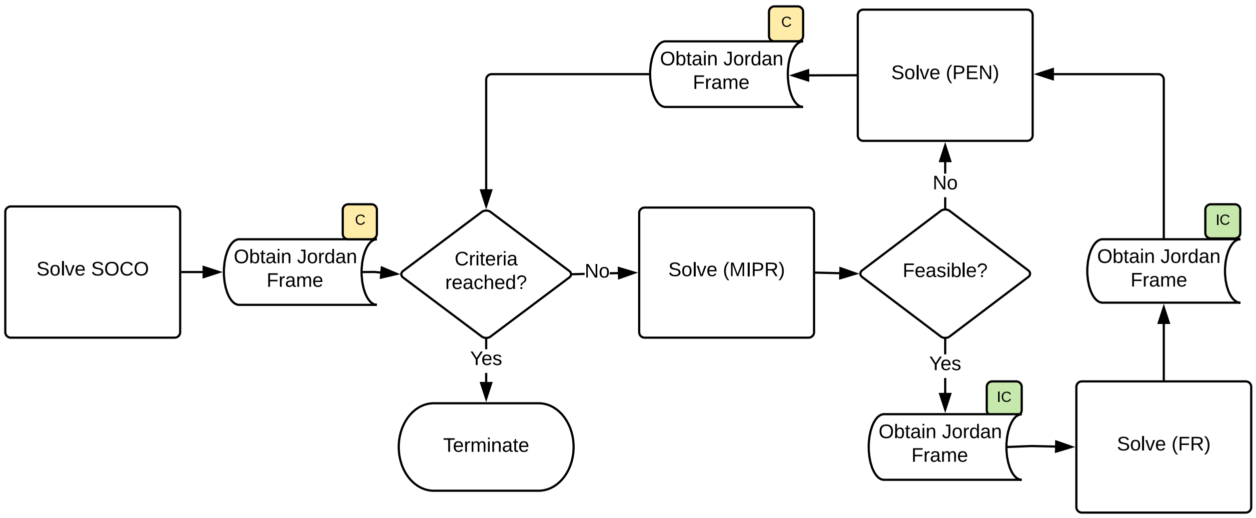


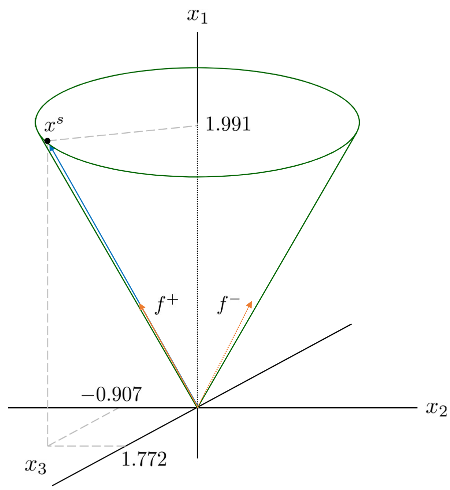
Figure 2: Flow of the primal rounding heuristic.

Algorithm 1 The primal rounding heuristic for MISOCO

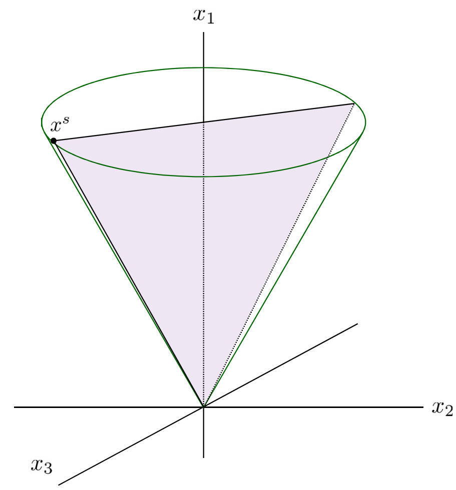
Input: A MISOCO instance (1),
maximum number of iterations t

Output: A feasible solution $\tilde{x}$ to MISOCO, if found

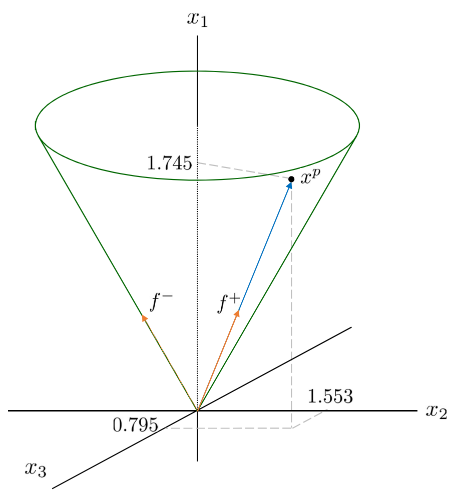
Set $\tilde{c} = \infty$, $\varphi = 0.5$
Solve the continuous relaxation of MISOCO, obtain its solution x^s
Add F^s to the Jordan frame pool
while $i \leq t$ **do**
  Solve (MIPR), obtain its solution x^* if exists
  if (MIPR) is feasible **then**
    Add F^* to the Jordan frame pool
    Solve (FR) using x^*, obtain its solution x^r
    Add F^r to the Jordan frame pool
    if $c^\top x^r \leq \tilde{c}$ **then**
      $\tilde{c} = c^\top x^r$, $\tilde{x} = x^r$
    $\varphi = \frac{1+\varphi}{2}$
  else
    $\varphi = \frac{\varphi}{2}$
  Solve the penalty problem (PEN), obtain its solution x^p
  Add F^p to the Jordan frame pool
  $i = i + 1$
return $\tilde{x}$



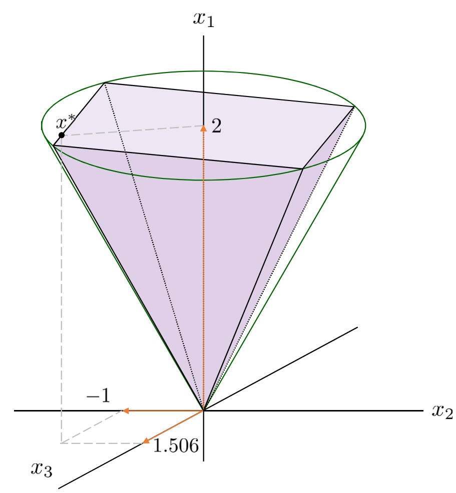
(a) Solution of the continuous relaxation, $x^s = (1.991, -0.907, 1.772)$.



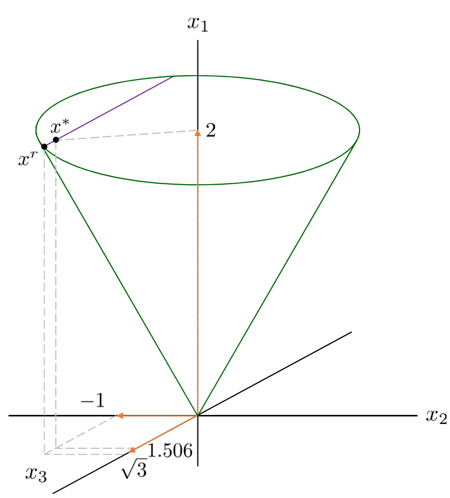
(b) Conic feasible region for (MIPR) (infeasible).



(c) Solution of the penalty problem (PEN), $x^p = (1.745, 1.553, 0.795)$.



(d) Conic feasible region for (MIPR), solution $x^* = (2, -1, 1.506)$.



(e) Solution of (FR), $x^r = (2, -1, \sqrt{3})$.

Figure 3: Steps of the primal rounding heuristic on the example problem.

(MIPR) is infeasible at this step. Supposing that we have budget for iteration, we continue. For illustration, see Figure 3b.

4. Solve the penalty problem (PEN) for $\varphi = 0.5$:

$$\begin{array}{rrcl} \text{minimize:} & \varphi(x_1 - x_3) + (1-\varphi)|x_3| & & \\ \text{subject to:} & 10\, x_1 + x_2 & = & 19, \\ & (x_1, x_2, x_3) & \in & \mathbb{L}^3. \end{array}$$

The optimal solution of (PEN) is $x^p = (1.745, 1.553, 0.795)$. For illustration, see Figure 3c.

5. Obtain the Jordan frames. The F matrix is updated:

$$F = \begin{bmatrix} F^+_{(1)} & F^-_{(1)} & F^+_{(2)} & F^-_{(2)} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ -0.228 & 0.228 & 0.445 & -0.445 \\ 0.445 & -0.445 & 0.228 & -0.228 \end{bmatrix}.$$

6. Solve the (MIPR) problem:

$$\begin{array}{rrcl} \text{minimize:} & 2x_1 + x_2 - 2x_3 & & \\ \text{subject to:} & 10\, x_1 + x_2 & = & 19, \\ & x_1 & = & 0.5\lambda_1 + 0.5\lambda_2 + 0.5\lambda_3 + 0.5\lambda_4, \\ & x_2 & = & -0.228\lambda_1 + 0.228\lambda_2 + 0.445\lambda_3 - 0.445\lambda_4, \\ & x_3 & = & 0.445\lambda_1 - 0.445\lambda_2 + 0.228\lambda_3 - 0.228\lambda_4, \\ & x_1 & \geq & 0, \\ & x_1, x_2 & \in & \mathbb{Z}, \\ & \lambda_1, \lambda_2, \lambda_3, \lambda_4 & \in & \mathbb{R}_+. \end{array}$$

The optimal solution of (MIPR) is $x^* = (2, -1, 1.506)$, the current upper bound is –0.012. For illustration, see Figure 3d.

7. Solve the (FR) problem:

$$\begin{array}{rrcl} \text{minimize:} & 2x_1 + x_2 - 2x_3 & & \\ \text{subject to:} & 10\, x_1 + x_2 & = & 19, \\ & x_1 & = & 2, \\ & x_2 & = & -1, \\ & (x_1, x_2, x_3) & \in & \mathbb{L}^3, \end{array}$$

The (FR) problem improves the solution and provides a feasible MISOCO solution $x^r = (2, --1, \sqrt{3})$ with objective value -0.464. For illustration, see Figure 3e.

8. Obtain Jordan frames. The Jordan frame is

$$F = \begin{bmatrix} 0.5 & 0.5 \\ -0.25 & 0.25 \\ 0.433 & -0.433 \end{bmatrix}.$$

9. Solve the (MIPR) problem:

$$\begin{array}{rrcl} \text{minimize:} & 2x_1 + x_2 - 2x_3 & & \\ \text{subject to:} & 10\, x_1 + x_2 & = & 19, \\ & x_1 & = & 0.5\lambda_1 + 0.5\lambda_2, \\ & x_2 & = & -0.25\lambda_1 + 0.25\lambda_2, \\ & x_3 & = & 0.433\lambda_1 - 0.433\lambda_2, \\ & x_1 & \geq & 0, \\ & x_1, x_2 & \in & \mathbb{Z}, \\ & \lambda_1, \lambda_2 & \in & \mathbb{R}_+. \end{array}$$

(MIPR) gives the same solution $x^* = (2, -1, \sqrt{3})$.

The relaxation objective for this problem is –0.4697, where the bound generated by the conic rounding heuristic is –0.4641. Despite the small gap, we cannot guarantee optimality in general. However, the solution provided by the heuristic is the unique optimal solution for this example.

3.1.2 Discussion

Despite following a simple logic, the primal rounding heuristic works well in practice. However, the performance depends highly on the feasible region of the problem. For problems where feasible solutions appear inside second-order cone (SOC), it is relatively easy to find Jordan frames where a feasible solution lies inside their convex combinations. On the other hand, if all the feasible solutions appear on, or close to the boundary of the SOCs, then we need to find the exact Jordan frame to obtain a feasible solution. Finding the exact Jordan frame is not always possible. In our experiments with CBLIB test problems, we observed that the primal rounding heuristic is not able to produce feasible solutions for stochastic service system design (`sssd`) problems. To provide an alternative to the primal rounding heuristic for problems where solutions are close to the boundary of SOCs, we propose the dual rounding heuristic in the following subsection.

3.2 The dual rounding heuristic

Using the relationship between the rounding problems and the original MISOCO instances, we can approach the problem of finding a feasible solution from the dual side. As shown in Figure 1, (D-DR) is a relaxation of the original MISOCO instance. The underlying idea for the dual rounding heuristic is to solve a series of (D-DR) problems with integrality constraints and (FR) problems.

The dual rounding heuristic starts with the solution of the continuous relaxation of the original instance. Next, the mixed-integer dual rounding problem (MIDR) is solved by using the Jordan frame matrix that was obtained from the relaxation solution, which is written as follows:

$$\begin{array}{rrcll} \text{minimize:} & c^\top x & & & \\ \text{subject to:} & Ax & = & b, & \\ & F^\top x & \geq & 0, & \\ & x_1^i & \geq & 0, & i \in 1, \dots, k \\ & x_j & \in & \mathbb{Z}, & j \in J \subseteq N. \end{array} \tag{MIDR}$$

If the solution is conic feasible, then the solution is an optimal solution to the MISOCO problem, and we terminate. Otherwise, Jordan frames obtained from the solution are added to the matrix F. Note that we can still obtain the Jordan frame that corresponds to the projection of the (MIDR) solution onto $\mathcal{K}$. Adding this Jordan frame to matrix F cuts off the current (MIDR) solution for the next iteration. By using the conic infeasible solution, we solve an (FR) problem by fixing integer variables. If the problem is feasible, then we obtained a feasible solution to MISOCO. We add the corresponding Jordan frame to F, and continue to the next iteration.

Figure 4 shows an overview of the dual rounding heuristic, and Algorithm 2 describes the dual rounding heuristic steps.

In theory, one can keep adding unique Jordan frames until an outer-approximation of the $\mathcal{K}$ to the desired precision is achieved because the second-order cones are self-dual and the constraint $F^\top x \geq 0$ provides a supporting hyperplane for $\mathcal{K}$. The set of Jordan frames provides a rough outer-approximation of the SOCs since we often have a limited number of Jordan frames as shown in our numerical results.

Because of its relationship to the original MISOCO problem, (MIDR) is feasible at each iteration of the heuristic for a feasible MISOCO problem. Therefore, if (MIDR) becomes infeasible at any iteration of the dual rounding heuristic, then the original MISOCO problem is infeasible. Moreover, if the solution to (MIDR) satisfies conic feasibility $x \in \mathcal{K}$, then it is an optimal solution for MISOCO.

One of the major benefits of the dual rounding heuristic is the detection of infeasible cases. The other benefit is that the dual rounding heuristic provides a lower bound for the MISOCO problem, thus providing a better gap for the optimality information. A lower bound is provided even if the heuristic fails to find a feasible solution. As shown in our numerical experiments, the dual rounding heuristic works well for certain

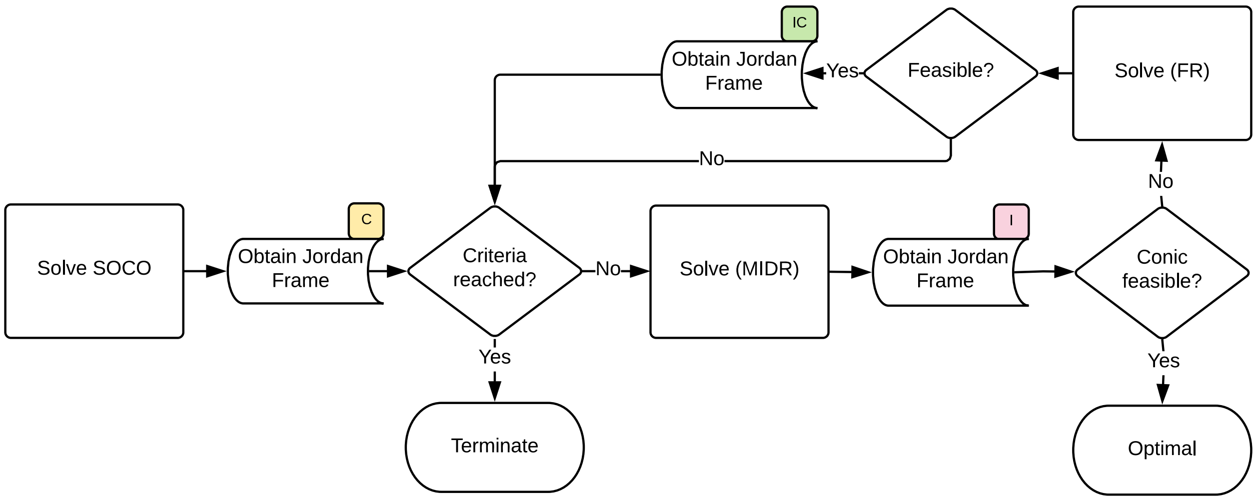


Figure 4: Flow of the dual rounding heuristic.

Algorithm 2 The dual rounding heuristic for MISOCO

Input: A MISOCO instance (1),
maximum number of iterations t
Output: A feasible solution $\tilde{x}$ to MISOCO if found, a global lower bound c_L
Set $\tilde{c} = \infty$
Solve the continuous relaxation of MISOCO, obtain its solution x^s, set $c_L = c^\top x^s$
Add F^s to the Jordan frame pool
while $i \leq t$ **do**
  Solve (MIDR), obtain solution x^*
  Add F^* to the Jordan frame pool
  if $c^\top x^* \geq c_L$ **then**
    $c_L = c^\top x^*$
  if $x^* \in \mathcal{K}$ **then**
    $\tilde{x} = x^*$, $\tilde{c} = c^\top x^*$
    Terminate with an optimal solution to MISOCO x^*.
  else
    Solve (FR) using x^*, obtain its solution x^r if exists
    if (FR) is feasible **then**
      Add F^r to the Jordan frame pool
      if $c^\top x^r \leq \tilde{c}$ **then**
        $\tilde{c} = c^\top x^r$, $\tilde{x} = x^r$
  if $c_L = \tilde{c}$ **then**
    Terminate with an optimal solution to MISOCO $\tilde{x}$.
  $i = i + 1$
return $\tilde{x}, c_L$

---

problem types and provides a good solution in a few iterations. The main difference between the dual rounding heuristic and the primal rounding heuristic is the existence of penalty problems. Since any solution of (MIDR) cuts off the current solution and effectively decreases the feasible region, there is no need to solve a separate penalty problem.

3.2.1 Example

In this subsection, we provide a numerical example to illustrate how the dual rounding heuristic works in practice. The steps of the dual rounding heuristic are illustrated in Figure 5.

Consider the following optimization model:

$$\begin{array}{rrcl} \text{minimize:} & -15x_2 - 8x_3 & & \\ \text{subject to:} & x_1 & = & 3, \\ & x_2, x_3 & \leq & 3, \\ & (x_1, x_2, x_3) & \in & \mathbb{L}^3, \\ & x_1, x_3 & \in & \mathbb{Z}. \end{array}$$

Following are the steps of the dual rounding heuristic. Numerical values are given up to three digits precision.

1. Solve the SOCO subproblem:

$$\begin{array}{rrcl} \text{minimize:} & -15x_2 - 8x_3 & & \\ \text{subject to:} & x_1 & = & 3, \\ & x_2, x_3 & \leq & 3, \\ & (x_1, x_2, x_3) & \in & \mathbb{L}^3. \end{array}$$

   The solution of the SOCO subproblem is $x^* = (3, 2.647, 1.412)$ with objective value –51. For illustration, see Figure 5a.

2. Obtain the Jordan frames. The F matrix is

$$F = \begin{bmatrix} 0.5 & 0.5 \\ 0.471 & -0.471 \\ 0.167 & -0.167 \end{bmatrix}.$$

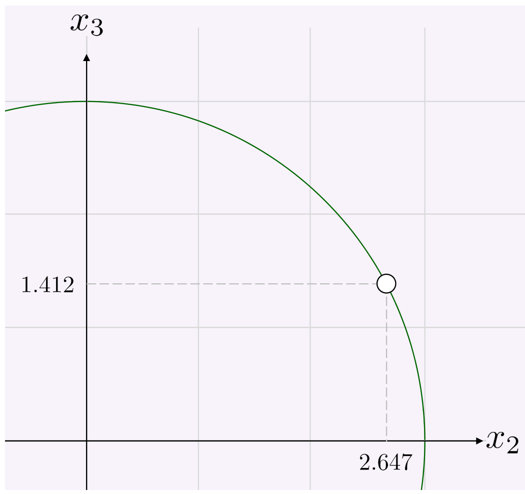
3. Solve the (MIDR) problem:

$$\begin{array}{rrcl} \text{minimize:} & -15x_2 - 8x_3 & & \\ \text{subject to:} & x_1 & = & 3, \\ & x_2, x_3 & \leq & 3, \\ & 0.5x_1 + 0.471x_2 + 0.167x_3 & \geq & 0, \\ & 0.5x_1 - 0.471x_2 - 0.167x_3 & \geq & 0, \\ & x_1 & \geq & 0, \\ & x_1, x_3 & \in & \mathbb{Z}. \end{array}$$

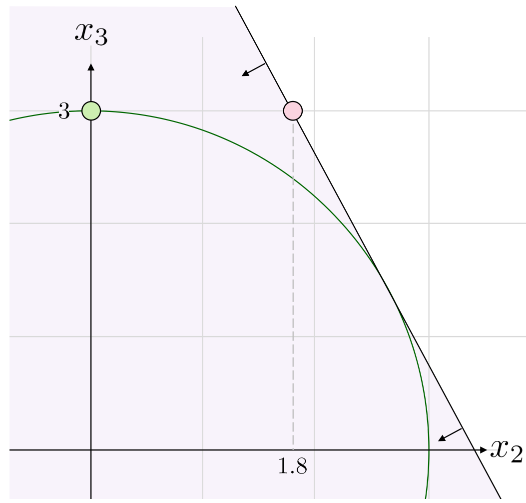
   (MIDR) provides a solution $\bar{x} = (3, 1.8, 3)$, which is conic infeasible. We fix the integer variables $x_1 = 3, x_3 = 3$ and solve the (FR) problem. The (FR) problem provides a conic and integer feasible solution $x^* = (3, 0, 3)$ with objective value –24. This is the best feasible solution for the MISOCO problem so far. For illustration, see Figure 5b.

4. Obtain Jordan frames from both the optimal solution of (MIDR) and the solution of (FR). The F matrix is

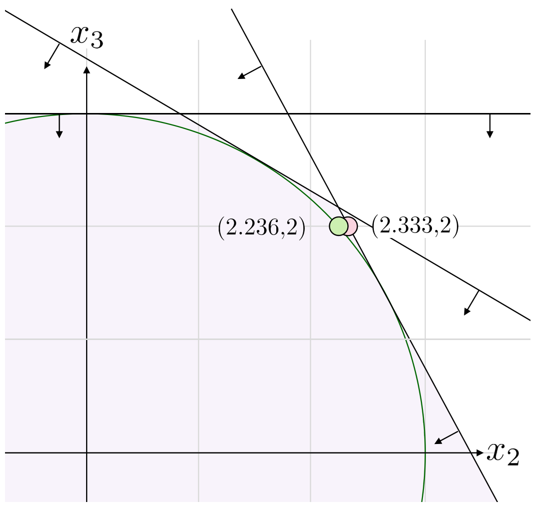
$$F = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.471 & -0.471 & 0.257 & -0.257 & 0 & 0 \\ 0.167 & -0.167 & 0.429 & -0.429 & 0.5 & -0.5 \end{bmatrix}.$$



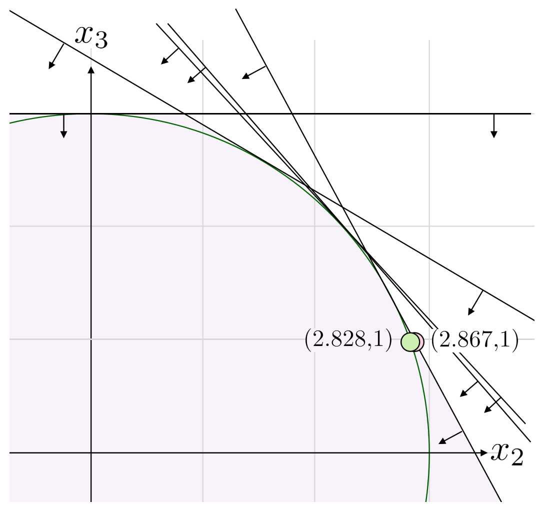
(a) Solution of the original continuous relaxation, $x^s = (3, 2.647, 1.412)$.



(b) Solutions obtained after the first iteration, $x^* = (3, 1.8, 3)$, $x^r = (3, 0, 3)$.



(c) Solutions obtained after the second iteration, $x^* = (3, 2.333, 2)$, $x^r = (3, 2.236, 2)$.



(d) Solutions obtained after the third iteration, $x^* = (3, 2.867, 1, 3)$, $x^r = (3, 2.828, 1)$.

Figure 5: Steps of the dual rounding heuristic on the sample problem for the cross section at $x_1 = 3$.

5. Solve the (MIDR) problem:

$$\begin{array}{lrcl} \text{minimize:} & -15x_2 - 8x_3 & & \\ \text{subject to:} & x_1 & = & 3, \\ & x_2, x_3 & \leq & 3, \\ & 0.5x_1 + 0.471x_2 + 0.167x_3 & \geq & 0, \\ & 0.5x_1 - 0.471x_2 - 0.167x_3 & \geq & 0, \\ & 0.5x_1 + 0.257x_2 + 0.429x_3 & \geq & 0, \\ & 0.5x_1 - 0.257x_2 - 0.429x_3 & \geq & 0, \\ & 0.5x_1 + 0.5x_3 & \geq & 0, \\ & 0.5x_1 - 0.5x_3 & \geq & 0, \\ & x_1 & \geq & 0, \\ & x_1, x_3 & \in & \mathbb{Z}. \end{array}$$

The solution of (MIDR) is $x^* = (3, 2.333, 2)$. We fix the integer variables $x_1 = 3, x_3 = 2$ and solve (FR). The (FR) provides a feasible solution $x^* = (3, 2.236, 2)$ with objective value –49.541. This is the best feasible solution for the MISOCO problem so far. For illustration see Figure 5c.

6. Obtain the Jordan frames.

$$F = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.471 & -0.471 & 0.257 & -0.257 & 0 & 0 & 0.379 & -0.379 & 0.372 & -0.372 \\ 0.167 & -0.167 & 0.429 & -0.429 & 0.5 & -0.5 & 0.325 & -0.325 & 0.333 & -0.333 \end{bmatrix}.$$

7. Solve the (MIDR) problem:

$$\begin{array}{lrcl} \text{minimize:} & -15x_2 - 8x_3 & & \\ \text{subject to:} & x_1 & = & 3, \\ & x_2, x_3 & \leq & 3, \\ & 0.5x_1 + 0.471x_2 + 0.167x_3 & \geq & 0, \\ & 0.5x_1 - 0.471x_2 - 0.167x_3 & \geq & 0, \\ & 0.5x_1 + 0.257x_2 + 0.429x_3 & \geq & 0, \\ & 0.5x_1 - 0.257x_2 - 0.429x_3 & \geq & 0, \\ & 0.5x_1 + 0.5x_3 & \geq & 0, \\ & 0.5x_1 - 0.5x_3 & \geq & 0, \\ & 0.5x_1 + 0.379x_2 + 0.325x_3 & \geq & 0, \\ & 0.5x_1 - 0.379x_2 - 0.325x_3 & \geq & 0, \\ & 0.5x_1 + 0.372x_2 + 0.333x_3 & \geq & 0, \\ & 0.5x_1 - 0.372x_2 - 0.333x_3 & \geq & 0, \\ & x_1 & \geq & 0, \\ & x_1, x_3 & \in & \mathbb{Z}. \end{array}$$

The solution of (MIDR) is $x^* = (3, 2.867, 1)$, which violates the conic constraint slightly. We fix the integer variables $x_1 = 3, x_3 = 1$ and solve (FR). The (FR) produces the solution $x^r = (3, 2.828, 1)$ with objective value –50.426. This is the best feasible solution for the MISOCO so far. For illustration, see Figure 5d.

Repeating the heuristic for one more iteration results in an (MIDR) solution of $x^* = (3, 2, 828, 1)$, which proves that the solution is the true optimal solution for the MISOCO. This example demonstrates a good case where only a few iterations provided the unique optimal solution. In general, there is no guarantee that a feasible solution can be found in a finite number of iterations.

3.2.2 Discussion

The dual rounding heuristic is useful for finding a feasible solution when the integer feasible solutions of MISOCO are close to the boundary of the SOCs. A pathological case occurs if there is a continuous region near the both sides of the boundary. Without (FR), infinitely many iterations are required to find a feasible solution in this case.

The dual rounding heuristic is similar to the primal rounding heuristic in terms of utilizing Jordan frames to generate feasible solutions. The main difference lies in where the solutions of the mixed-integer rounding problems appear. If feasible, (MIPR) produces a feasible solution to MISOCO. On the other hand, if MISOCO is feasible, then (MIDR) is always feasible, but its solution is likely to be infeasible for MISOCO because of violated conic constraints. These two different heuristics are useful for different cases. The primal rounding heuristic works better when feasible solutions are inside the SOCs, whereas it fails for cases where the feasible solutions are close to, or lie on the boundary of the SOCs. The dual rounding heuristic works better when feasible solutions are close to the boundary of SOCs, but fails when there is a continuous feasible region alongside a cone.

In the next subsection we present the primal-dual rounding heuristic, where the iteration budget is shared equally between the primal rounding and the dual rounding without needing penalty problems.

3.3 The primal-dual rounding heuristic

As we discussed in previous subsections, the primal rounding heuristic and the dual rounding heuristic are useful for different types of feasible regions. For a problem with known structure, one can decide which heuristic to deploy. However, one often does not have enough knowledge about the feasible region, making the decision of which heuristic to use. In this subsection, we propose a combination of two heuristics. This heuristic is called the *primal-dual rounding heuristic* and consists of components of both heuristics.

The primal-dual rounding heuristic starts with the solution of the SOCO problem and its Jordan frames. Then, the (MIPR) problem is solved using the frames available in the pool. If it is feasible, then an (FR) step follows to improve the solution. The generated solution is fed to the set of Jordan frames, and the best solution is updated. Then, whether the (MIPR) is feasible or not, the (MIDR) problem is solved. The (MIDR) problem is guaranteed to be always feasible unless the original problem is infeasible; hence the solution can be added to the Jordan frame pool. If the solution to (MIDR) is conic feasible, it means that we found an optimal solution to the original MISOCO problem. If not, then we use the solution in an (FR) step to see if the (MIDR) can lead to a feasible solution. If (FR) provides a feasible solution, then we add the corresponding Jordan frame to the pool. We iterate by solving (MIPR) until we reach a predefined iteration limit.

Figure 6 shows an overview of the primal-dual rounding heuristic, and Algorithm 3 describes the steps of the primal-dual rounding heuristic.

Although the primal-dual rounding heuristic is a combination of both of the primal and dual rounding heuristics that were introduced earlier, there are some differences in terms of the flow. The most significant difference between the primal rounding and the primal-dual rounding heuristic is how we generate new Jordan frames. A penalty problem is solved in every iteration of the primal rounding heuristic to generate new frames. This means that we need to solve two SOCOs in every iteration if we have a feasible solution. For the primal-dual rounding heuristic, we benefit from the frames generated by the (MIDR) solutions. Another advantage is that there is no need to solve a separate penalty problem to generate new frames. However, whereas the the primal rounding heuristic does not provide a lower bound, the dual rounding heuristic provides a lower bound at every iteration. By combining these two heuristics, we aim to provide a feasible solution and a better lower bound at the same time.

Algorithm 3 The primal-dual rounding heuristic for MISOCO

Input: A MISOCO instance (1),
maximum number of iterations t

Output: A feasible solution $\tilde{x}$ to MISOCO if found, a global lower bound c_L

Set $\tilde{c} = \infty$
Solve the continuous relaxation of MISOCO, obtain its solution x^s, set $c_L = c^\top x^s$
Add F^s to the Jordan frame pool
while $i \leq t$ **do**
    Solve (MIPR), obtain its solution x^* if it exists
    if (MIPR) feasible **then**
        Add F^* to the Jordan frame pool
        Solve (FR) using x^*, obtain its solution x^r
        Add F^r to the Jordan frame pool
        if $c^\top x^r \leq \tilde{c}$ **then**
            $\tilde{c} = c^\top x^r$, $\tilde{x} = x^r$
    Solve (MIDR), obtain solution x^*
    Add F^* to the Jordan frame pool
    if $x^* \in \mathcal{K}$ **then**
        $\tilde{c} = c^\top x^*, \tilde{x} = x^*$
        Terminate with an optimal solution to MISOCO.
    if $c^\top x^* \geq c_L$ **then**
        $c_L = c^\top x^*$
    Solve (FR) using x^*, obtain its solution x^r if it exists
    if (FR) feasible **then**
        Add F^r to the Jordan frame pool
        if $c^\top x^r < \tilde{c}$ **then**
            $\tilde{c} = c^\top x^r$, $\tilde{x} = x^r$
    if $c_L = \tilde{c}$ **then**
        Terminate with an optimal solution to MISOCO
    $i = i + 1$
return $\tilde{x}, c_L$

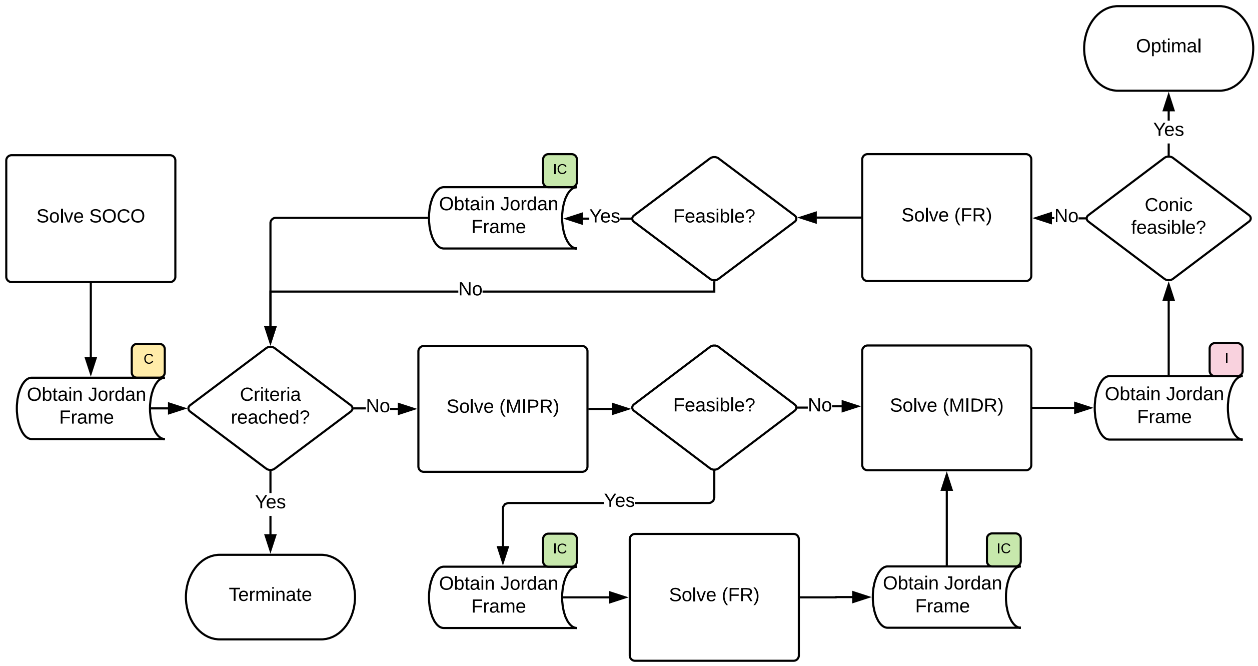


Figure 6: Flow of the primal-dual rounding heuristic.

3.4 Hybrid strategy

The primal-dual rounding heuristic allocates budget equally among primal and dual rounding heuristics. Moreover, primal rounding uses only Jordan frames that are generated by the dual rounding heuristic, specifically by (MIDR) and dual (FR) problems. In our experiments we noticed that the primal rounding heuristic provides a feasible solution to the majority of the test problems within a few iterations. Therefore, we present a hybrid strategy (HS) as a combination of two heuristics, where a few iterations are pure primal rounding heuristic and the remaining are dual rounding heuristic only. They do not interact, except we carry Jordan frames obtained in the primal side to the dual rounding heuristic and start it from the existing Jordan frames. Although this sounds intuitive, the benefit of carrying existing Jordan frames might be controversial, because it sometimes changes the frames obtained from the dual side significantly.

4 Numerical results

4.1 Implementation and test set

For computational experiments, we implemented all heuristics in MATLAB 2014b[2]. The SOCO problems are solved with MOSEK 7.1, and the MILO problems are solved with CPLEX 12.7 via their respective MATLAB API. Problems are solved on an AMD Opteron 2.0 GHz with 8GB of memory.

One of the most used test sets for MISOCO instances is CBLIB [26]. We tested our heuristics on all MISOCO instances in this test set; however, because the memory is limited to 8 GB, only 1322 out of 1754 test problems are included in the numerical experiments. We also converted 6 convex binary quadratic optimization problems from the QPLIB test set [27] into MISOCO using MOSEK and applied our heuristics on these problems as well. The problem types and corresponding cone sizes are shown in Table 1.

Table 1 shows a huge variety of problems, where some problems have a few relatively big cones (QPLIB) and some problems have multiple small cones (`pp` and `turbine`). All problems are converted to the

[2]Please contact the corresponding author to receive a copy of the source code.

		Variables		Integers		Cones		Cone sizes	
Pr. Types	#P	Min	Max	Min	Max	Min	Max	Min	Max
ck	90	611	3271	25	75	10	20	27	77
classical	399	146	356	20	50	1	1	21	51
estein	9	125	246	9	18	9	18	3	3
pp	3	72	702	10	100	10	100	3	3
robust	400	198	468	21	51	2	2	22	52
shortfall	400	194	464	21	51	2	2	21	51
sssd	14	273	785	72	264	12	24	3	3
turbine	7	121	512	11	56	25	119	3	3
QPLIB	6	3033	13538	20	400	1	1	802	4502
Summary	1328	72	13538	9	400	1	119	3	4502

Table 1: Details of the problem test set.

standard MISOCO form using MOSEK. After conversion, only three problems in the test set (`turbine07`, `turbine07_aniso` and `turbine54`) have integer variables that appear in multiple-dimensional cones. All of these problems have 11 integers and all of them are leading variable in their cones. No problems in the standard form have integer variables as an in-cone variable, but it is likely that integer variables are related to in-cone variables in some of the test problems.

4.2 Efficiency of the heuristics

First, we experimented with the heuristics to see how often they provide a feasible solution and how many iterations it takes to produce a feasible solution. Table 2 shows how many iterations it takes for heuristics to generate the first feasible solution for MISOCO problems. We limited the number of MILO problems to be solved by 10. For the hybrid strategy, we allocate three out of 10 MILO problem budget to primal rounding heuristic and the remaining to dual rounding heuristic.

The most spectacular results in Table 2 are the percentage of problems where the primal rounding heuristic provided a solution and the number of iterations to generate them. The primal rounding heuristic fails to find a feasible solution for only 19 out of 1329 problems. Out of the 98.5% of problems that the primal rounding heuristic provided a solution for, a feasible solution is obtained for 48% and 96% problems in only one and two iterations, respectively. This is a remarkable result for a primal heuristic in general. Moreover, the hybrid strategy fails only for three problems in total.

In terms of feasibility, the dual rounding heuristic provides a feasible solution for 78% of all test problems, where it provided a solution in one iteration for 33% and in two iterations for 44% of all test problems. The primal-dual rounding heuristic provides a feasible solution for 80% of all test problems with 65% feasible in two iterations. The primal-dual rounding heuristic is expected to perform between the primal rounding heuristic and the dual rounding heuristic, since the iteration budget is essentially halved between these two except minor differences.

As discussed after introducing the primal rounding heuristic, an interesting result can be seen from the Table 2 for `sssd` problems. The primal heuristic fails to find a feasible solution in 10 iterations for all problems in this set. Coincidentally, the dual rounding heuristic and also the primal-dual rounding heuristic provide a feasible solution in only one iteration for all problems in this class.

One can conclude that there is little chance for the primal rounding heuristic to provide a solution after two iterations. However, the dual rounding and the primal-dual rounding heuristics may provide a solution in subsequent iterations.

Overall, there is only one problem, `robust_50_51`, where all four heuristics failed to provide a solution with the given iteration limit. The primal rounding heuristic, the dual rounding heuristic, and the primal-

		# Iters											
Heur	P.Type	1	2	3	4	5	6	7	8	9	10	Failed	Total
P		647	628	28	2	3			1		1	18	1328
	ck	90											90
	classical	5	393	1									399
	estein	9											9
	pp		2	1									3
	robust	136	231	24	2	3			1		1	2	400
	shortfall	400											400
	sssd											14	14
	turbine	2	2	2								1	7
	QPLIB	5										1	6
D		445	146	73	80	50	57	52	46	50	44	285	1328
	ck	10	15	13	8	8	5	6	9	3		13	90
	classical	140	42	19	30	15	19	16	9	18	14	77	399
	estein	9											9
	pp		2					1					3
	robust	120	40	24	33	16	16	19	20	17	13	82	400
	shortfall	142	45	17	8	11	17	10	8	12	17	113	400
	sssd	14											14
	turbine	4	2		1								7
	QPLIB	6											6
PD		647	230	8	68		36	2	53	1	23	260	1328
	ck	90											90
	classical	5	138		42		18		30		15	151	399
	estein	9											9
	pp			3									3
	robust	136	73	5	25		18	2	23	1	8	109	400
	shortfall	400											400
	sssd		14										14
	turbine	2	4		1								7
	QPLIB	5	1										6
HS		647	629	27	16	2	2			2		3	1328
	ck	90											90
	classical	5	393	1									399
	estein	9											9
	pp		3										3
	robust	136	231	24	2	2				2		3	400
	shortfall	400											400
	sssd				13		1						14
	turbine	2	2	2			1						7
	QPLIB	5			1								6

Table 2: Number of iterations where the first feasible solution to MISOCO is reported.

dual rounding heuristic are able to find a solution for this problem in 12 iterations, 19 iterations, and 19 iterations, respectively. To sum up, all heuristics, especially the primal rounding heuristic, work well in practice in terms of finding solutions in a few iterations. This is a remarkable performance for heuristics of this type.

4.3 Quality of the provided solutions

In this part, we evaluate the quality of the provided solution at the termination. Notice that all heuristics may terminate early if the objective value of the generated solution is equal to the global lower bound, hence proving its optimality. This is more likely for the dual and the primal-dual rounding heuristics, since they keep generating new lower bounds for the problem over iterations.

Table 3 shows detailed performance of the heuristics over problem types. In the table, number of problems (#P), number of problems solved (#S), average reported gap (RG), average true gap (TG), average number of iterations to reach the first feasible solution (IFS), average number of MILOs solved (#MILO), average number of SOCOs solved (#SOCO), and the total number of instances where the best solution is provided by the primal (FR) problem, the dual (FR) problem, and the (MIDR) problem are presented. It is apparent that the reported gap of the primal rounding heuristic is significantly higher compared to its true gap from the optimal solution for some problem types, `estein`, `turbine`, and `QPLIB`. This is mainly due to the gap between the solution of the continuous relaxation and the solution of the MISOCO problem.

Another interesting result is that the dual rounding heuristic provides an optimal solution for `ck` problems, whenever feasible. Since (FR) fails to find a solution in most of the cases, it is very likely that integer variables are directly related to in-cone variables.

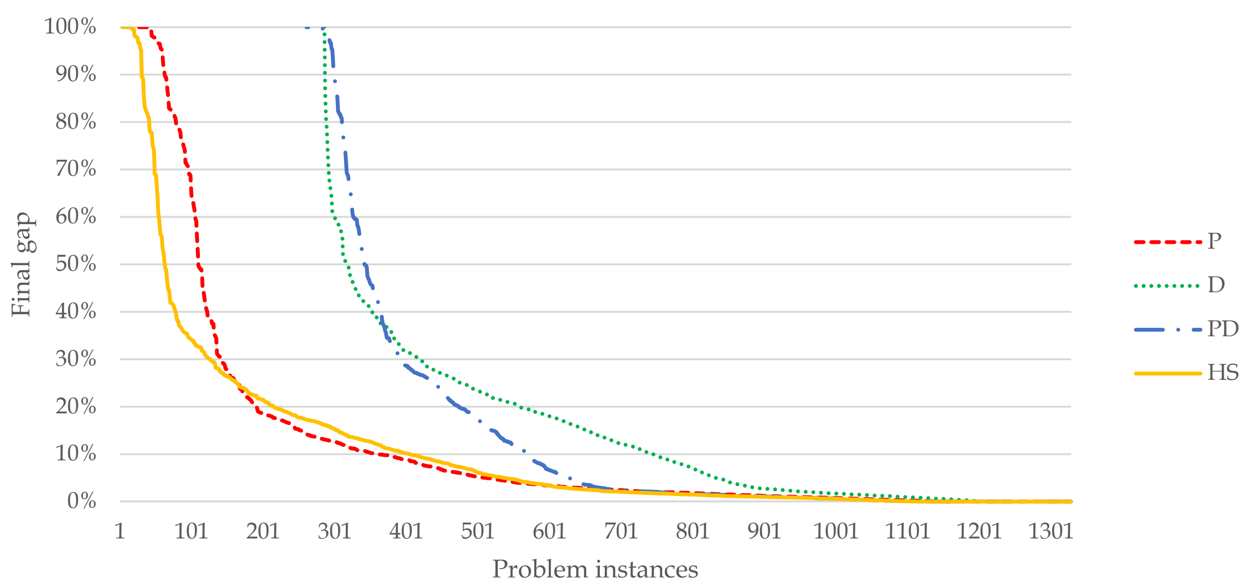


Figure 7: Comparison of gaps to the true optimal across heuristics.

See Figure 7 for a representation of the gaps to the true optimal solution across all problems. Problems are sorted based on the gap provided by the heuristic, and the plots are shifted to show failed problems. Also see Figure 8 for a comparison of gap to the true optimal with the first feasible solutions versus the number of iterations to first feasible solution. We can conclude that the primal rounding heuristic provides smaller gaps in fewer steps despite failing to provide a solution for certain problem types. The dual rounding heuristic usually takes more iterations, but the gap provided is usually close to the true optimal solution, unlike the primal rounding heuristic where many solutions provide a gap close to 100% at the first feasible solution. The optimal strategy is likely a combination of primal and dual heuristics on average, but it is up

P.Type	Heur	#P	#S	RG	TG	IFS	#MILO	#SOCO	FR-P	FR-D	MIDR
ck											
	P	90	90	75.36%	71.67%	1.00	10.00	20.00	90	0	0
	D	90	77	0.00%	0.00%	4.18	5.02	4.22	0	5	72
	PD	90	90	35.94%	35.90%	1.00	7.36	6.79	37	3	50
	HS	90	90	38.59%	38.54%	1.00	8.11	9.57	41	2	47
classical											
	P	399	399	10.28%	8.56%	1.99	9.80	18.43	399	0	0
	D	399	322	20.86%	19.52%	3.34	10.00	10.00	0	322	0
	PD	399	248	16.44%	15.03%	3.82	9.91	6.84	117	131	0
	HS	399	399	15.04%	13.42%	1.99	9.83	10.71	366	33	0
estein											
	P	9	9	99.48%	0.09%	1.00	10.00	20.00	9	0	0
	D	9	9	0.15%	0.00%	1.00	9.11	9.11	0	9	0
	PD	9	9	1.69%	0.00%	1.00	10.00	10.00	2	7	0
	HS	9	9	1.26%	0.04%	1.00	9.56	11.56	3	6	0
pp											
	P	3	3	0.22%	0.13%	2.33	10.00	19.00	3	0	0
	D	3	3	1.27%	1.26%	3.67	9.33	9.00	0	2	1
	PD	3	3	0.00%	0.00%	3.00	10.00	8.67	2	0	1
	HS	3	3	33.33%	33.34%	2.00	10.00	11.00	3	0	0
robust											
	P	400	398	10.84%	10.34%	1.79	7.65	14.19	398	0	0
	D	400	318	20.30%	19.91%	3.64	9.97	9.97	0	318	0
	PD	400	291	19.43%	19.07%	2.72	8.18	5.92	215	76	0
	HS	400	397	11.59%	11.09%	1.78	7.64	8.52	370	27	0
shortfall											
	P	400	400	1.65%	1.49%	1.00	9.72	19.44	400	0	0
	D	400	287	1.70%	1.56%	3.08	10.00	10.00	0	287	0
	PD	400	400	1.41%	1.25%	1.00	9.76	9.76	340	60	0
	HS	400	400	1.44%	1.28%	1.00	9.73	11.71	365	35	0
sssd											
	P	14	0	-	-	-	10.00	10.00	0	0	0
	D	14	14	0.00%	0.00%	1.00	8.29	7.29	0	6	8
	PD	14	14	0.22%	0.06%	2.00	10.00	8.86	7	7	0
	HS	14	14	3.33%	0.95%	4.14	10.00	8.93	7	7	0
turbine											
	P	7	6	43.79%	4.51%	2.00	10.00	17.43	6	0	0
	D	7	7	0.00%	0.00%	1.71	4.00	3.57	0	5	2
	PD	7	7	0.00%	0.00%	2.00	4.86	3.57	3	2	2
	HS	7	7	0.00%	0.00%	2.57	5.71	5.86	4	1	2
QPLIB											
	P	6	5	90.15%	73.91%	1.00	10.00	18.17	5	0	0
	D	6	6	70.40%	46.53%	1.00	10.00	10.00	0	6	0
	PD	6	6	70.42%	46.51%	1.17	10.00	8.83	1	5	0
	HS	6	6	72.69%	51.02%	1.50	10.00	11.00	1	5	0

Table 3: Detailed performance of the heuristics on problem types.

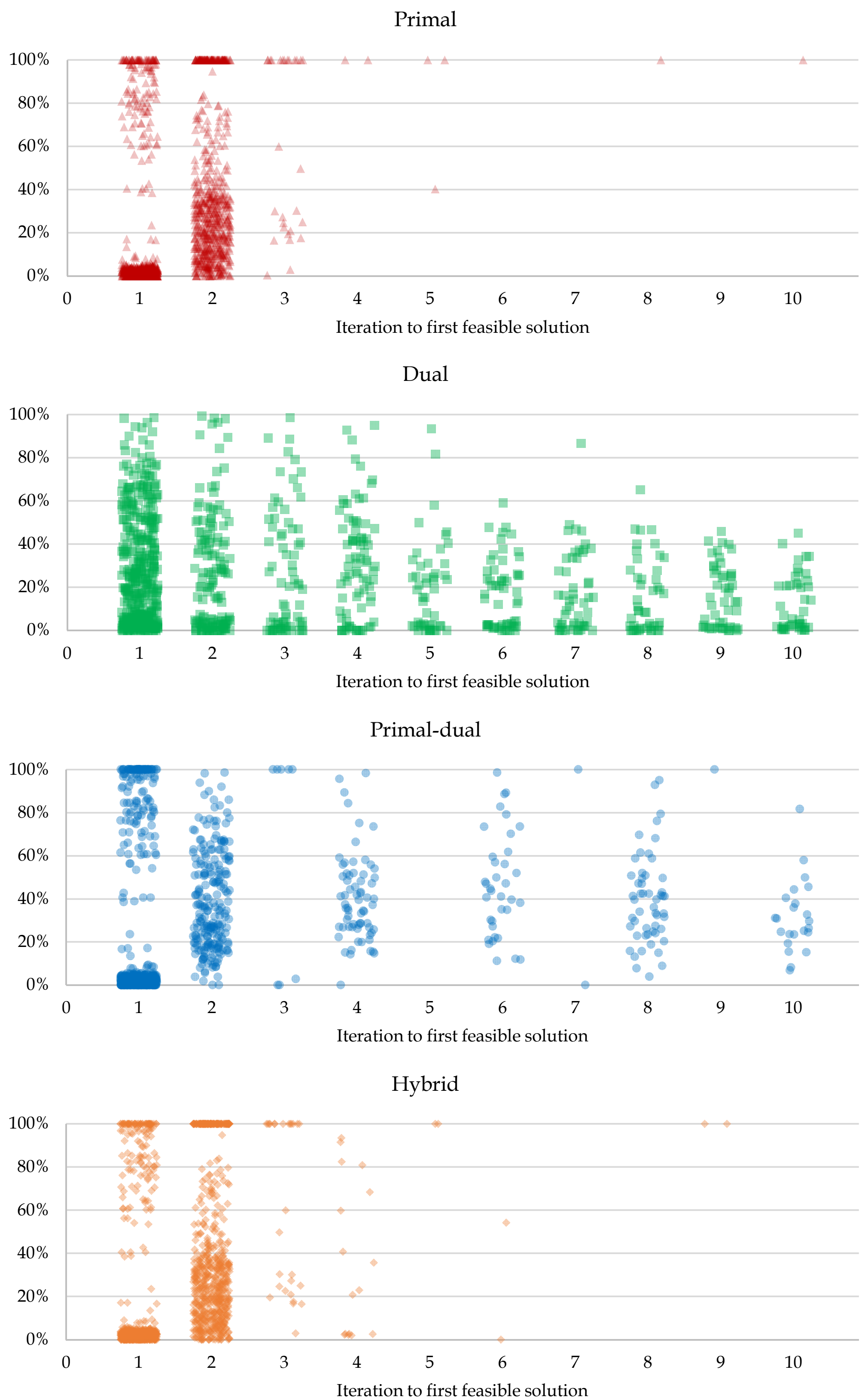


Figure 8: Gap to the true optimal versus the number of iterations to first feasible solution for each heuristic.

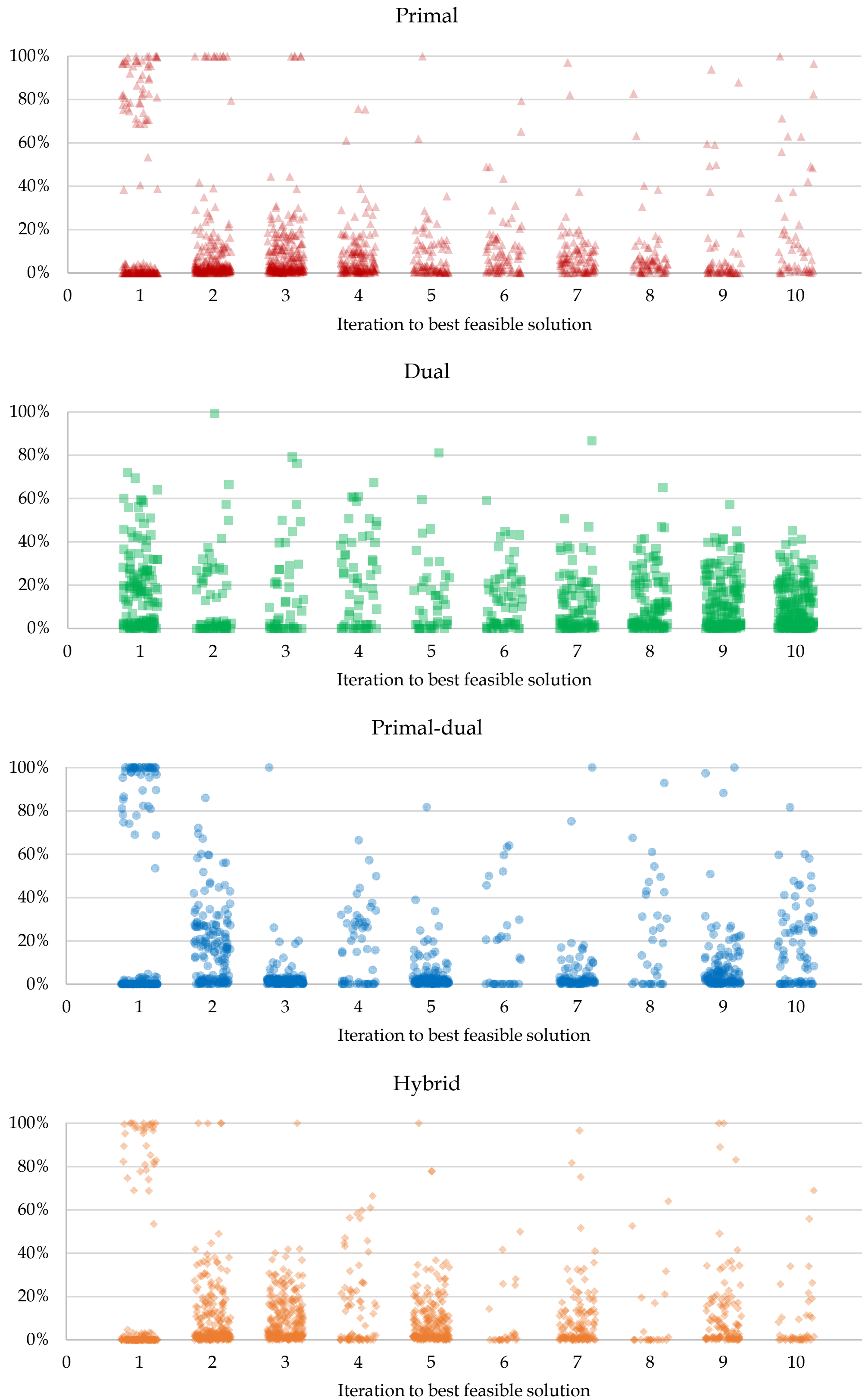


Figure 9: Gap to the true optimal versus the number of iterations to best feasible solution for each heuristic.

to the user's knowledge to find the best combination. See Figure 9 for a comparison of the gap to the true optimal with the best solution obtained versus the number of iterations to get the best feasible solution. It is apparent that having multiple iterations helps reduce the gap significantly. The solution provided by the heuristic keeps getting better as we keep iterating. The HS dominates other heuristics in terms of final gaps.

In some cases, the heuristic stops early and returns the solution obtained if the objective value is equal to the best known lower bound. In these cases, we prove the optimality of the solution. Table 4 shows the number of instances and iterations where the true optimal solution is obtained. Note that there are instances where an optimal solution is found but cannot be proved because the lower bound is not sufficient, which is a common case for the primal rounding heuristic. These instances are not included in this table.

		# Iters										
Heur	P.Type	1	2	3	4	5	6	7	8	9	10	Total
P		107	13	4	3	1		2			2	132
	classical	5	2	1	2						2	12
	robust	93	8	3		1		2				107
	shortfall	9	3		1							13
D		14	19	13	11	9	8	12	13	7	6	112
	ck	10	15	13	8	8	5	6	9	3		77
	estein	2	2		2	1						7
	pp								1			1
	robust						1	2		2	5	10
	sssd						2	4	3	2	1	12
	turbine	2	2		1							5
PD		107	14	13	16	2	12	3	8		9	184
	ck		10		15	1	12		8		8	54
	classical	5						1				6
	estein		2									2
	pp										1	1
	robust	91		9		1		2				103
	shortfall	9		4								13
	turbine	2	2		1							5
HS		109	16	3	18	1	13	6	15	2	7	190
	ck				15		12	1	15	1	7	51
	classical	5	2	1				1				9
	estein		1		1							2
	robust	93	10	2				4				109
	shortfall	9	3			1				1		14
	turbine	2			2		1					5

Table 4: Number of instances where an optimal solution is found.

The primal-dual heuristic is capable of finding more optimal solutions because primal and dual rounding heuristics work together to provide upper and lower bounds, respectively. This result verifies our intuition about the hybrid strategy.

4.4 Effect of iterations on solution quality

One can terminate the heuristics whenever a feasible solution is obtained. However, taking more iterations often provides a much better solution.

See Figure 10 for a scatter plot of the gap of the first feasible solution and the final solution after a maximum of 10 MILO problems.

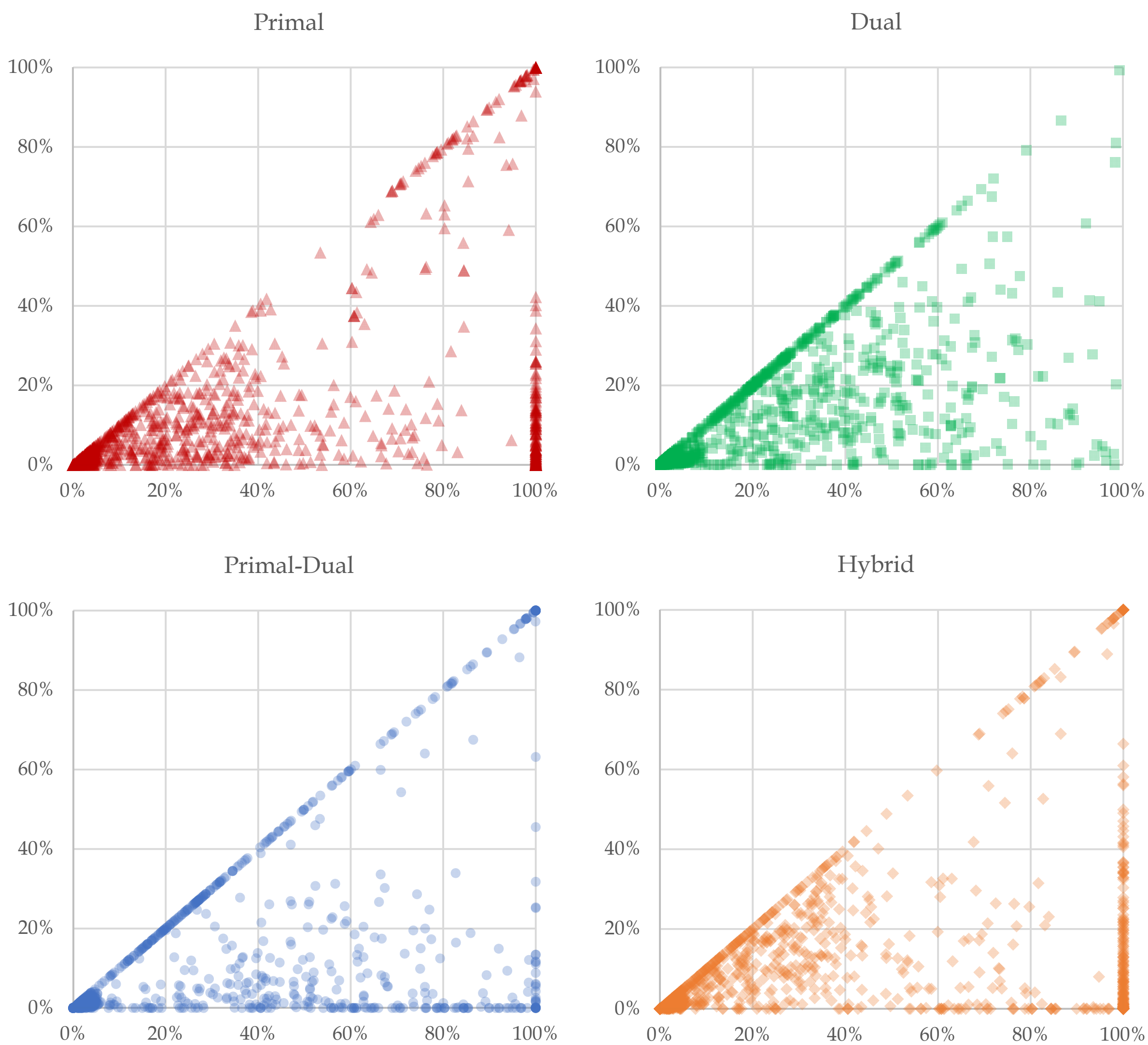


Figure 10: Final gap to the true optimal (vertical axis) versus the first gap (horizontal axis) for each instance.

These plots show that having more iterations helps the primal rounding heuristic very significantly, especially for cases where the initial gap is around 100%. The effect of multiple iterations is very significant for the primal-dual rounding heuristic, where the gap to true optimality decreases from the 60—100% interval to 0%.

The solutions obtained after having even a few iterations are significantly better. Even for the problems where the first feasible solution is obtained in the first iteration, the gap to the true optimal is 33%, 64%, 51%, and 36% better when we obtain the best known solution in the second iteration for primal, dual, primal-dual rounding heuristics, and the hybrid strategy, respectively. Having more iterations gradually improves the gap compared to the first feasible solution. On average, the primal rounding heuristics provide 54% better gap, while the number is 45% for the dual rounding and 54% for the primal-dual rounding. Overall, having multiple iterations results in an average of 52% better gap at return. These numbers show the significance of running the heuristic for a few iterations.

5 Conclusions and future work

In this paper, we presented four novel rounding heuristics for MISOCO problems. The proposed heuristics are specific to MISOCO and provide major contributions to the existing body of the research. As the first of its kind, the conic rounding heuristics complement the theoretical developments in the MISOCO area and their implementations provide a cost-efficient and novel strategy for finding feasible solutions.

Based on our computational results on the CBLIB and QPLIB test sets, HS finds a feasible solution for almost all test problems. This is a significant result for a heuristic method, it would be significant even for linear optimization. Overall, using a budget of 10 MILO problems, at least one of the heuristics found a feasible solution to 1327 out of 1328 total problems.

This study is an important step to open new research directions for future studies. Due to the close relationship between SOCO and semi-definite optimization (SDO), these heuristics can also be applied to SDO problems. However, it might be expensive to obtain eigenvectors for larger and dense matrices for SDO problems. Therefore, more work is needed to develop efficiently computable versions of these heuristics for SDO problems.

Acknowledgment

The authors acknowledge the support of the Air Force Office of Scientific Research grant #FA9550-15-1-0222 for the development of this research.

References

[1] T. Achterberg. *Constraint Integer Programming*. PhD thesis, Technische Universität Berlin, 2008.

[2] T. Achterberg and T. Berthold. Improving the feasibility pump. *Discrete Optimization*, 4(1):77–86, 2007.

[3] T. Achterberg, T. Berthold, T. Koch, and K. Wolter. Constraint integer programming: A new approach to integrate CP and MIP. In L. Perron and M. A. Trick, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 6–20. Springer, 2008.

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

[5] E. D. Andersen and K. D. Andersen. The MOSEK optimization software. *EKA Consulting ApS, Denmark*, 2000.

[6] K. Andersen and A. N. Jensen. Intersection cuts for mixed integer conic quadratic sets. In M. Goemans and J. Correa, editors, *Integer Programming and Combinatorial Optimization*, volume 7801, pages 37–48. Springer, 2013.

[7] A. Atamtürk and V. Narayanan. Conic mixed-integer rounding cuts. *Mathematical Programming*, 122 (1):1–20, 2010.

[8] A. Atamtürk and V. Narayanan. Lifting for conic mixed-integer programming. *Mathematical Programming*, 126(2):351–363, 2011.

[9] P. Belotti, J. C. Góez, I. Pólik, T. K. Ralphs, and T. Terlaky. On families of quadratic surfaces having fixed intersections with two hyperplanes. *Discrete Applied Mathematics*, 161(16):2778–2793, 2013.

[10] P. Belotti, J. C. Góez, I. Pólik, T. K. Ralphs, and T. Terlaky. A conic representation of the convex hull of disjunctive sets and conic cuts for integer second order cone optimization. In M. Al-Baali, L. Grandinetti, and A. Purnama, editors, *Numerical Analysis and Optimization*, pages 1–35. Springer, 2015.

[11] P. Belotti, J. C. Góez, I. Pólik, T. K. Ralphs, and T. Terlaky. A complete characterization of disjunctive conic cuts for mixed integer second order cone optimization. *Discrete Optimization*, 24:3–31, 2017.

[12] H. Y. Benson and Ü. Sağlam. Mixed-integer second-order cone programming: A survey. In H. Topaloglu, J. C. Smith, and H. J. Greenberg, editors, *Theory Driven by Influential Applications*, pages 13–36. INFORMS, 2013.

[13] L. Bertacco, M. Fischetti, and A. Lodi. A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization*, 4(1):63–76, 2007.

[14] T. Berthold. Primal Heuristics for Mixed Integer Programs. Master's thesis, Technische Universität Berlin, 2006.

[15] T. Berthold. *Heuristic Algorithms in Global MINLP Solvers*. PhD thesis, Technische Universität Berlin, 2014.

[16] T. Berthold, S. Heinz, and S. Vigerske. Extending a CIP framework to solve MIQCPs. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 427–444. Springer, 2012.

[17] P. Bonami and J. P. Gonçalves. Heuristics for convex mixed integer nonlinear programs. *Computational Optimization and Applications*, 51(2):729–747, 2012.

[18] P. Bonami, M. Kilinç, and J. Linderoth. Algorithms and software for convex mixed integer nonlinear programs. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 1–39. Springer, 2012.

[19] S. B. Çay, J. C. Góez, and T. Terlaky. Effects of disjunctive conic cuts within a branch and conic cut algorithm to solve asset allocation problems. Technical Report 16T-005, Lehigh University, 2016.

[20] S. B. Çay, I. Pólik, and T. Terlaky. Warm-start of interior point methods for second order cone optimization via rounding over optimal Jordan frames. Technical Report 17T-006, Lehigh University, 2017.

[21] M. T. Çezik and G. Iyengar. Cuts for mixed 0-1 conic programming. *Mathematical Programming*, 104 (1):179–202, 2005.

[22] S. Drewes. *Mixed Integer Second Order Cone Programming*. PhD thesis, Technische Universität Darmstadt, 2009.

[23] J. Faraut and A. Korányi. *Analysis on Symmetric Cones*. Oxford Science Publications, 1994.

[24] FICO™ Xpress Optimization Suite. *XPRESS-Optimizer, Reference manual*, Fair Isaac Corporation, 2009.

[25] M. Fischetti, F. Glover, and A. Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, 2005.

[26] H. A. Friberg. CBLIB 2014: a benchmark library for conic mixed-integer and continuous optimization. *Mathematical Programming Computation*, 8(2):191–214, 2016.

[27] F. Furini, E. Traversi, P. Belotti, A. Frangioni, A. Gleixner, N. Gould, L. Liberti, A. Lodi, R. Misener, H. Mittelmann, N. Sahinidis, S. Vigerske, and A. Wiegele. QPLIB: A library of quadratic programming instances. Technical report, February 2017. URL http://www.optimization-online.org/DB_HTML/2017/02/5846.html.

[28] K. Ghobadi, H. R. Ghaffari, D. M. Aleman, D. A. Jaffray, and M. Ruschin. Automated treatment planning for a dedicated multi-source intracranial radiosurgery treatment unit using projected gradient and grassfire algorithms. *Medical Physics*, 39(6):3134–3141, 2012.

[29] Z. Gu, E. Rothberg, and R. E. Bixby. Gurobi optimizer reference manual, version 6.0. *Gurobi Optimization Inc., Houston, USA*, 2014.

[30] IBM ILOG CPLEX. *V12. 1: Users Manual for CPLEX*, 2009.

[31] F. Kılınç-Karzan. On minimal valid inequalities for mixed integer conic programs. *Mathematics of Operations Research*, 41(2):477–510, 2015.

[32] S. Modaresi, M. R. Kılınç, and J. P. Vielma. Intersection cuts for nonlinear integer programming: Convexification techniques for structured sets. *Mathematical Programming*, 155(1-2):575–611, 2016.

[33] M. Ç. Pınar. Mixed-integer second-order cone programming for lower hedging of American contingent claims in incomplete markets. *Optimization Letters*, 7(1):63–78, 2013.

[34] I. Pólik and J. C. Góez. Rounding solutions in SOCP. ICCOPT, Lisbon, Portugal, 2013.

[35] T. Terlaky and I. Pólik. Parametric second order cone optimization and its applications: Challenges and perspectives. NSF Grant Proposal, 2010. Lehigh University.

[36] S. Yıldız and G. Cornuéjols. Disjunctive cuts for cross-sections of the second-order cone. *Operations Research Letters*, 43(4):432–437, 2015.