ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, IL 60439

Exploiting Second-Order Cone Structure for Global Optimization

Ashutosh Mahajan and Todd Munson

Mathematics and Computer Science Division

Preprint ANL/MCS-P1801-1010

October 28, 2010

Contents

Exploiting Second-Order Cone Structure for Global Optimization

Ashutosh Mahajan* Todd Munson†

October 28, 2010

Abstract

Identifying and exploiting classes of nonconvex constraints whose feasible region is convex after branching can reduce the time to compute global solutions for nonlinear optimization problems. We develop techniques for identifying quadratic and nonlinear constraints whose feasible region can be represented as the union of a finite number of second-order cones, and we provide necessary and sufficient conditions for some reformulations. We then construct a library of small instances where these reformulations are applicable. Comparing our method to general-purpose solvers, we observe several orders of magnitude improvement in performance.

1 Introduction

Consider a constrained optimization problem of the form,

$$\begin{aligned}&\min_{x\in\mathbb{R}^n} f(x)\\ &\text{s.t. } p_i(x)\le 0, \quad i=1,\dots,m,\end{aligned} \tag{P}$$

where $n, m \in \mathbb{N}$, $f : \mathbb{R}^n \to \mathbb{R}$, and $p_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \dots, m$ are given. When the feasible region is a convex set and f is a convex function, the problem is relatively easy to solve because any local solution is a global solution. Convexity of the feasible region is guaranteed if p_i is a convex function for each $i = 1, \dots, m$. Conditions sufficient for ensuring convexity have been described by Boyd and Vandenberghe [2004], amongst others.

If the feasible region is not convex, a local solution may not be a global solution and we must search the feasible region for a global solution. Branch-and-bound methods are typically used for such a search. Global optimization solvers usually approximate nonconvex constraints by convex relaxations. A secant approximation [Tawarmalani and Sahinidis, 2004] is used to linearize concave functions, for example, while McCormick inequalities are used [Al-Khayyal et al., 1995] for bilinear functions. Convex quadratic relaxations can also be constructed by using the approach of Androulakis et al. [1995]. The tightness of the relaxation depends on the bounds on the variables; this fact is used to generate stronger relaxations as the feasible region is explored. In particular, after solving a convex relaxation, we can branch on a variable $x_0 \le k \vee x_0 \ge k$ for some $k \in \mathbb{R}$ and use the tightened bounds to strengthen the linear relaxations. Instead of branching, we can generate valid inequalities [Saxena et al., 2008; Belotti, 2010] by solving the so-called cut-generating linear program.

In this paper, we identify classes of nonconvex constraints whose feasible region is convex after branching. The main idea underlying our branching scheme is to identify subdomains where the feasible region can be represented as the union of a finite number of second-order cones. A second-order cone is a convex set defined as

$$S = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^1 \mid y \le \|x\|_2\},$$

where $\|.\|_2$ denotes the two norm. Alizadeh and Goldfarb [2003] describe the theory and applications of second-order cone programming. They also describe methods to reformulate some special convex problems, such as the harmonic mean of positive affine functions, using second-order cones. In contrast, we partition a given nonconvex region so that each subdomain is a second-order cone. The simplest such region is the set

$$T = \{x_0 \in \mathbb{R} \mid k - x_0^2 \le 0\},$$

where $k > 0$ is a given parameter. Even though $k - x_0^2$ is not a convex function and T is not a convex set, if we branch on the disjunction $x_0 \le 0 \vee x_0 \ge 0$, then we obtain two convex sets:

$$T_0 = \{x_0 \in \mathbb{R} \mid x_0 \le -\sqrt{k}\}, \quad T_1 = \{x_0 \in \mathbb{R} \mid x_0 \ge \sqrt{k}\}.$$

*Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439; e-mail: mahajan@mcs.anl.gov
†Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439; e-mail: tmunson@mcs.anl.gov

A generalization in higher dimensions is the set

$$S = \{x \in \mathbb{R}^n \mid \|\tilde{A}x + \tilde{b}\|_2^2 \leq (\tilde{c}^T x + \tilde{d})^2\},$$

where $\tilde{A} \in \mathbb{R}^n \times \mathbb{R}^n, \tilde{b} \in \mathbb{R}^n, \tilde{c} \in \mathbb{R}^n$, and $\tilde{d} \in \mathbb{R}$ are given. In particular, S is the union of two second-order cones

$$S_0 = \{x \in \mathbb{R}^n \mid \|\tilde{A}x + \tilde{b}\|_2 \leq (\tilde{c}^T x + \tilde{d})\}, \quad S_1 = \{x \in \mathbb{R}^n \mid \|\tilde{A}x + \tilde{b}\|_2 \leq -(\tilde{c}^T x + \tilde{d})\}.$$

We determine, in Section 2, the conditions for expressing a given constraint in the form of the set S for a general class of nonlinear constraints containing a quadratic term and, in Section 3, for factorable constraints, where more than two second-order cones may be needed.

Once this structure is identified, we need not create any relaxations for the constraint since branching results in convex second-order cone constraints. When we have branched on all such constraints, the feasible region is convex, and we need not branch further. Such a branching scheme can significantly reduce the size of the branch-and-bound tree required to compute a global solution when compared to the tree generated by existing solvers using linear relaxations. Moreover, our approach is applicable when the bounds on the variables are not finite. This approach, however, can be used only on a restricted set of functions. The computational results in Section 4 demonstrate that this method can outperform existing methods by orders of magnitude on small instances that are difficult to solve with spatial branching.

2 Nonlinear Constraints

We consider nonlinear constraints of the form

$$x^T Ax + c^T x + d + \sum_{i=1}^{k} g_i(x) \leq 0, \tag{1}$$

where $A \in \mathbb{R}^{n \times n}$ is a symmetric matrix, $c \in \mathbb{R}^n$, $d \in \mathbb{R}$, and $g_i(x) : \mathbb{R}^n \to \mathbb{R}$ is a convex function for each $i = 1, \ldots, k$. If A is positive semidefinite, all eigenvalues of A are nonnegative. Then the feasible region is a convex set, and we need not reformulate the constraint.

Therefore, we assume A is not positive semidefinite and we want to reformulate constraint (1) using second-order cones. We first assume $g_i(x)$ is a nonnegative, convex function and $\sqrt{g_i(x)}$ is a convex function. Adding variables and constraints produces the equivalent augmented system,

$$\begin{array}{rcl} x^T Ax + c^T x + d + \sum_{i=1}^k z_i^2 & \leq & 0, \\ \sqrt{g_i(x)} & \leq & z_i, \quad i = 1, \ldots, k, \end{array} \tag{2}$$

where $z \in \mathbb{R}^k$ are auxiliary variables. In particular, the assumptions made on $g_i(x)$ guarantee the new constraints are convex. Moreover, the introduction of auxiliary variables does not increase the number of negative eigenvalues in the quadratic constraint.

Theorem 2.1. *Assume $g_i(x)$ is a nonnegative function for each $i = 1, \ldots, k$. If $x^* \in \mathbb{R}^n$ is feasible for constraint (1) then there exists $z^* \in \mathbb{R}^k$ such that (x^*, z^*) is feasible for constraints (2). Conversely, if (x^*, z^*) is feasible for constraints (2), then x^* is feasible for constraint (1).*

Proof. Given $x^* \in \mathbb{R}^n$ such that $x^{*T} Ax^* + c^T x^* + d + \sum_{i=1}^k g_i(x^*) \leq 0$. By assumption, $g_i(x^*) \geq 0$. Let $z_i^* = \sqrt{g_i(x^*)}, i = 1, \ldots, k$. Then (x^*, z^*) is feasible for constraints (2).

Conversely, if $(x^*, z^*) \in \mathbb{R}^{n+k}$ is feasible for constraints (2), then $g_i(x^*) \leq z_i^{*2}, i = 1, \ldots, k$. Summing the following $k + 1$ inequalities

$$x^{*T} Ax^* + c^T x^* + d + \sum_{i=1}^{k} z_i^{*2} \leq 0$$

$$g_i(x^*) - z_i^{*2} \leq 0 \quad i = 1, \ldots, k,$$

shows that x^* is feasible for constraint (1). □

We now consider two examples where the functions $g_i(x)$ in constraint (1) are not zero.

Example 1: We can reformulate the constraint

$$x^T Ax + c^T x + \sum_{i=1}^{k} \alpha_i^{r_i^T x} + d \leq 0,$$

when $\alpha_i > 0, i = 1, \ldots, k$, as

$$\begin{aligned} x^T Ax + c^T x + \sum_{i=1}^{k} x_{n+i}^2 + d &\leq 0 \\ \alpha_i^{\frac{1}{2} r_i^T x} - x_{n+i} &\leq 0, \quad i = 1, \ldots, k, \end{aligned}$$

since $\alpha_i^{\frac{1}{2} r_i^T x}$ is convex. ■

Example 2: Consider the constraint

$$x^T Ax + c^T x + \sum_{i=1}^{k} \left| r_i^T x \right|^{p_i} + d \leq 0,$$

where $k \in \mathbb{N}$ and $p_i \geq 2$. We can reformulate this constraint as

$$\begin{aligned} x^T Ax + c^T x + \sum_{i=1}^{k} x_{n+i}^2 + d &\leq 0, \\ \left| r_i^T x \right|^{\frac{p_i}{2}} &\leq x_{n+i}, \quad i = 1, \ldots, k, \end{aligned}$$

since $|r_i^T x|^{\frac{p_i}{2}}$ is convex. ■

The only nonconvex constraint in our reformulation is the quadratic constraint. Therefore, consider a quadratic constraint in the general form

$$x^T Ax + c^T x + d \leq 0, \tag{3}$$

where $A \in \mathbb{R}^{n \times n}$ is a symmetric matrix that is not positive semidefinite. When possible, we want to recover a second-order cone constraint

$$\|\tilde{A}x + \tilde{b}\|_2 \leq \tilde{c}^T x + \tilde{d}.$$

By squaring both sides of this inequality, we obtain a quadratic constraint with the Hessian matrix

$$\tilde{A}^T \tilde{A} - \tilde{c}\tilde{c}^T,$$

that is, a positive semidefinite matrix with a rank-one update. This matrix can have at most one negative eigenvalue [e.g., Thompson, 1976]. Therefore, if A has more than one negative eigenvalue, then the constraint cannot be transformed into a second-order cone constraint. A having exactly one negative eigenvalue, however, does not imply that the constraint can be transformed into a second-order cone constraint; additional conditions are required. We now detail the construction used to identify constraints that can be represented as second-order cone constraints. We then prove that the conditions are necessary.

2.1 Construction

We first perform an eigenvalue decomposition of A into QDQ^T, where Q is an orthogonal matrix and D is a diagonal matrix containing the eigenvalues. Since A is symmetric, all its eigenvalues are real [Wilkinson, 1965, p. 25]. We then introduce a diagonal scaling matrix R and a diagonal matrix E with entries from $\{0, -1, 1\}$ such that $D = RER$:

$$R_{ii} = \begin{cases} \sqrt{|D_{ii}|} & \text{if } D_{ii} \neq 0 \\ 1 & \text{if } D_{ii} = 0 \end{cases}, \quad E_{ii} = \begin{cases} -1 & \text{if } D_{ii} < 0 \\ 0 & \text{if } D_{ii} = 0 \\ 1 & \text{if } D_{ii} > 0. \end{cases}$$

The original constraint (3) is now equivalent to the pair of constraints

$$\begin{aligned} y^T Ey + b^T y + d &\leq 0, \\ y &= RQ^T x, \end{aligned}$$

where $b = R^{-1}Q^T c$.

We then split the indices of y into four disjoint sets:

1. Let I_+ denote the indices where E has a positive entry, $I_+ = \{i \mid E_{ii} > 0\}$.
2. Let I_- denote the indices where E has a negative entry, $I_- = \{i \mid E_{ii} < 0\}$.
3. Let I_0 denote the indices where E has a zero entry and the corresponding entry in the linear term b is nonzero, $I_0 = \{i \mid E_{ii} = 0, b_i \neq 0\}$.
4. Let $I_{\bar{0}}$ denote the indices where E has a zero entry and the corresponding entry in the linear term b is zero, $I_{\bar{0}} = \{i \mid E_{ii} = 0, b_i = 0\}$.

For the purposes of the construction we ignore the set $I_{\bar{0}}$, since they have no impact on the constraint. Constraint (3) is therefore equivalent to the pair of constraints

$$\begin{aligned} \sum_{i \in I_+} \left(y_i^2 + b_i y_i\right) - \sum_{i \in I_-} \left(y_i^2 - b_i y_i\right) + \sum_{i \in I_0} (b_i y_i) + d \leq 0, \\ y = RQ^T x. \end{aligned}$$

We now complete the squares to produce the equivalent constraints

$$\begin{aligned} \sum_{i \in I_+} \left(y_i + \frac{b_i}{2}\right)^2 - \sum_{i \in I_-} \left(y_i - \frac{b_i}{2}\right)^2 + \sum_{i \in I_0} (b_i y_i) + d + \frac{\sum_{i \in I_-} b_i^2 - \sum_{i \in I_+} b_i^2}{4} \leq 0, \\ y = RQ^T x. \end{aligned} \tag{4}$$

We immediately note that if $|I_-| = 0$, then constraint (3) has a convex feasible region. Otherwise, if $|I_-| > 1$, then constraint (3) cannot be written as a second-order cone constraint since the original matrix has more than one negative eigenvalue. Therefore, we need consider only the case where $|I_-| = 1$. Assuming without loss of generality that $I_- = \{1\}$, we now rearrange the terms to produce the equivalent constraints:

$$\begin{aligned} \sum_{i \in I_+} \left(y_i + \frac{b_i}{2}\right)^2 + \sum_{i \in I_0} (b_i y_i) + d + \frac{\sum_{i \in I_-} b_1^2 - \sum_{i \in I_+} b_i^2}{4} \leq \left(y_1 - \frac{b_1}{2}\right)^2, \\ y = RQ^T x. \end{aligned} \tag{5}$$

If $|I_0| = 0$ and the constant $h = d + \frac{\sum_{i \in I_-} b_i^2 - \sum_{i \in I_+} b_i^2}{4}$ is nonnegative, then the constraint can be written as

$$\left\| \frac{E_+ \left(y + \frac{b}{2}\right)}{\sqrt{h}} \right\|_2 \leq \left| y_1 - \frac{b_1}{2} \right|, \tag{6}$$

where E_+ is a diagonal $n \times n$ matrix, with $E^+_{ii} = E_{ii}$ if $i \in I_+$ and $E^+_{ii} = 0$ otherwise. The region feasible to the above constraint is a union of two second-order cones. We can obtain a second-order cone formulation by branching on the disjunction $y_1 \leq \frac{b_1}{2} \vee y_1 \geq \frac{b_1}{2}$. The following two systems of inequalities are used in place of constraint (3) in each of the branches, respectively:

$$\begin{aligned} \left\| \frac{E_+ \left(y + \frac{b}{2}\right)}{\sqrt{h}} \right\|_2 \leq \frac{b_1}{2} - y_1, \\ y = RQ^T x; \end{aligned} \qquad \begin{aligned} \left\| \frac{E_+ \left(y + \frac{b}{2}\right)}{\sqrt{h}} \right\|_2 \leq y_1 - \frac{b_1}{2}, \\ y = RQ^T x. \end{aligned}$$

We can also rewrite (6) as a system of inequalities with convex functions and an additional binary variable,

$$\begin{aligned} \left\| \frac{E_+ \left(y + \frac{b}{2}\right)}{\sqrt{h}} \right\|_2 &\leq \frac{b_1}{2} - y_1 + M y_b \\ \left\| \frac{E_+ \left(y + \frac{b}{2}\right)}{\sqrt{h}} \right\|_2 &\leq y_1 - \frac{b_1}{2} + M(1 - y_b), \\ y_b &\in \{0, 1\}, \end{aligned} \tag{7}$$

where M is suitably large, and then optimize over such a system by using a solver for convex mixed-integer nonlinear optimization problems.

2.2 Necessary Conditions

We identify three conditions for reformulation:

(C1) $|I_-| = 1$,

(C2) $|I_0| = 0$, and

(C3) $h = d + \frac{\sum_{i\in I_-} b_i^2 - \sum_{i\in I_+} b_i^2}{4}$ is nonnegative,

Our construction shows that if these conditions are satisfied, then the feasible region of constraint (3) is a union of two second-order cones. We now show these three conditions are necessary. If the constraint is not redundant and at least one of these conditions is violated, then the feasible region of constraint (3) is not a union of a finite number of second-order cones.

The outline of our proofs follows a general scheme. For each case, we find three feasible points for constraint (3), x^1, x^2, and x^3, such that the pairwise midpoints, $x^{12} = \frac{1}{2}(x^1 + x^2)$, $x^{23} = \frac{1}{2}(x^2 + x^3)$, and $x^{31} = \frac{1}{2}(x^3 + x^1)$, are infeasible. Therefore, x^1 and x^2 do not belong to a convex set. We find similar results for the pairs x^2, x^3 and x^3, x^1. We then conclude that the feasible region of constraint (3) is not a union of two convex sets.

To complete the proofs, we eliminate the additional y variables from constraint (4) to obtain the equivalent quadratic constraint in the space of original variables,

$$\sum_{i\in I_+}\left(R_{ii}Q_i^T x + \frac{b_i}{2}\right)^2 - \sum_{i\in I_-}\left(R_{ii}Q_i^T x - \frac{b_i}{2}\right)^2 + \sum_{i\in I_0\cup I_{\bar{0}}}\left(b_i R_{ii} Q_i^T x\right) + h \le 0, \tag{8}$$

where Q_i is the i^{th} column of the orthogonal matrix Q, $b = R^{-1}Q^T c$ and $h = d + \frac{\sum_{i\in I_-} b_i^2 - \sum_{i\in I_+} b_i^2}{4}$. In the following proofs we define

$$f(x) = \sum_{i\in I_+}\left(R_{ii}Q_i^T x + \frac{b_i}{2}\right)^2 - \sum_{i\in I_-}\left(R_{ii}Q_i^T x - \frac{b_i}{2}\right)^2 + \sum_{i\in I_0\cup I_{\bar{0}}}\left(b_i R_{ii} Q_i^T x\right) + h$$

and denote the set of feasible solutions as $F = \{x \mid f(x) \le 0\}$. We first preclude the trivial case when constraint (3) or, equivalently, constraint (8) is redundant.

Lemma 2.1. *Given a quadratic constraint in the form of constraint (8). If $|I_+| = |I_0| = 0$ and $h \le 0$, then constraint (8) is redundant.*

Proof. The constraint can be written as

$$h \le 0 \le \sum_{i\in I_-}\left(R_{ii}Q_i^T x - \frac{b_i}{2}\right)^2,$$

which is true for all $x \in \mathbb{R}^n$. $\square$

Theorem 2.2. *Given a quadratic constraint in the form of constraint (8). If $|I_-| > 0$ and $|I_0| > 0$, then F is not a union of two convex sets.*

Proof. Let $\bar{i}_- \in I_-$ and $\bar{i}_0 \in I_0$. Consider the system

$$RQ^T x = \bar{y}.$$

Let x^1 be the unique solution of the system when

$$\bar{y}_i = \begin{cases} 1 + \frac{b_i}{2}, & i = \bar{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \bar{i}_-, \\ \frac{-b_i}{2}, & i \in I_+, \\ \frac{1-h}{b_i}, & i = \bar{i}_0, \\ 0, & \text{otherwise.} \end{cases}$$

Let x^2 be the unique solution of the system when

$$\bar{y}_i = \begin{cases} -1 + \frac{b_i}{2}, & i = \bar{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \bar{i}_-, \\ \frac{-b_i}{2}, & i \in I_+, \\ \frac{1-h}{b_i}, & i = \bar{i}_0, \\ 0, & \text{otherwise.} \end{cases}$$

Let x^3 be the unique solution of the system when

$$\bar{y}_i = \begin{cases} \frac{b_i}{2}, & i \in I_-, \\ \frac{-b_i}{2}, & i \in I_+, \\ \frac{-h}{b_i}, & i = \bar{i}_0, \\ 0, & \text{otherwise.} \end{cases}$$

All three points x^1, x^2, x^3 are feasible since $f(x^1) = f(x^2) = f(x^3) = 0$. Consider the midpoint $x^{12} = \frac{1}{2}(x^1 + x^2)$. Then,

$$RQ^T x^{12} = \begin{cases} \frac{b_i}{2}, & i \in I_-, \\ \frac{-b_i}{2}, & i \in I_+, \\ \frac{1-h}{b_i}, & i = \bar{i}_0, \\ 0, & \text{otherwise,} \end{cases}$$

and $f(x^{12}) = 1$. Similarly, we can show $f(x^{23}) = f(x^{31}) = \frac{1}{4}$. Thus x^{12}, x^{23}, x^{31} are not feasible and F is not a union of two convex sets. □

Theorem 2.3. *Given a quadratic constraint in the form of constraint (8). If the constraint is not redundant and $|I_-| > 0$, $|I_0| = 0$, and $h < 0$, then F is not a union of two convex sets.*

Proof. Since the constraint is not redundant, it follows from Lemma 2.1 that $|I_+| > 0$. Let $\bar{i}_- \in I_-$ and $\bar{i}_+ \in I_+$. Consider the system

$$RQ^T x = \bar{y}.$$

Let x^1 be the unique solution of the system when

$$\bar{y}_i = \begin{cases} \frac{b_i}{2}, & i \in I_-, \\ \sqrt{-h} - \frac{b_i}{2}, & i = \bar{i}_+, \\ \frac{-b_i}{2}, & i \in I_+ \setminus \bar{i}_+, \\ 0, & \text{otherwise.} \end{cases}$$

Let x^2 be the unique solution of the system when

$$\bar{y}_i = \begin{cases} 1 + \frac{b_i}{2}, & i = \bar{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \bar{i}_-, \\ \sqrt{1-h} - \frac{b_i}{2}, & i = \bar{i}_+, \\ \frac{-b_i}{2}, & i \in I_+ \setminus \bar{i}_+, \\ 0, & \text{otherwise.} \end{cases}$$

Let x^3 be the unique solution of the system when

$$\bar{i}_i = \begin{cases} -1 + \frac{b_i}{2}, & i = \bar{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \bar{i}_-, \\ \sqrt{1-h} - \frac{b_i}{2}, & i = \bar{i}_+, \\ \frac{-b_i}{2}, & i \in I_+ \setminus \bar{i}_+, \\ 0, & \text{otherwise.} \end{cases}$$

All three points x^1, x^2, x^3 are feasible since $f(x^1) = f(x^2) = f(x^3) = 0$. Their midpoints x^{12}, x^{23}, x^{31}, however, are infeasible because $f(x^{12}) = f(x^{31}) = \frac{h + \sqrt{-h}\sqrt{1-h}}{2} > 0$ and $f(x^{23}) = 1$. Therefore, F is not a union of two convex sets. □

Theorem 2.4. *Given a quadratic constraint in the form of constraint (8). If the constraint is not redundant and $|I_-| > 1$, $|I_0| = 0$ and $h \geq 0$, then F is not a union of two convex sets.*

Proof. Let $\bar{i}_- \in I_-$ and $\tilde{i}_- \in I_-$ with $\bar{i}_- \neq \tilde{i}_i$. Consider the system

$$RQ^T x = \bar{y}.$$

Since the constraint is not redundant, it follows from Lemma 2.1 that either $h > 0$ or I_+ is not empty. We prove the theorem for these two cases separately.

(i) Suppose $h > 0$. Let x^1 be the unique solution of the system when

$$\bar{y}_i = \begin{cases} \sqrt{h} + \frac{b_i}{2}, & i = \bar{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \bar{i}_-, \\ \frac{-b_i}{2}, & i \in I_+, \\ 0, & \text{otherwise.} \end{cases}$$

Let x^2 be the unique solution of the system when

$$\bar{y}_i = \begin{cases} -\sqrt{h} + \frac{b_i}{2}, & i = \bar{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \bar{i}_-, \\ \frac{-b_i}{2}, & i \in I_+, \\ 0, & \text{otherwise.} \end{cases}$$

Let x^3 be the unique solution of the system when

$$\bar{y}_i = \begin{cases} \sqrt{h} + \frac{b_i}{2}, & i = \tilde{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \tilde{i}_-, \\ \frac{-b_i}{2}, & i \in I_+, \\ 0, & \text{otherwise.} \end{cases}$$

All three points x^1, x^2, x^3 are feasible since $f(x^1) = f(x^2) = f(x^3) = 0$. Their midpoints x^{12}, x^{23}, x^{31}, however, are infeasible because $f(x^{12}) = h > 0$ and $f(x^{23}) = f(x^{31}) = \frac{h}{2} > 0$. Therefore, F is not a union of two convex sets.

(ii) Suppose $h = 0$ and $|I_+| > 0$. Let $\bar{i}_+ \in I_+$. Let x^1 be the unique solution of the system when

$$\bar{y}_i = \begin{cases} 1 + \frac{b_i}{2}, & i = \bar{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \bar{i}_-, \\ 1 - \frac{b_i}{2}, & i = \bar{i}_+, \\ \frac{-b_i}{2}, & i \in I_+ \setminus \bar{i}_+, \\ 0, & \text{otherwise.} \end{cases}$$

Let x^2 be the unique solution of the system when

$$\bar{y}_i = \begin{cases} -1 + \frac{b_i}{2}, & i = \bar{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \bar{i}_-, \\ 1 - \frac{b_i}{2}, & i = \bar{i}_+, \\ \frac{-b_i}{2}, & i \in I_+ \setminus \bar{i}_+, \\ 0, & \text{otherwise.} \end{cases}$$

Let x^3 be the unique solution of the system when

$$\bar{y}_i = \begin{cases} 1 + \frac{b_i}{2}, & i = \tilde{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \tilde{i}_-, \\ 1 - \frac{b_i}{2}, & i = \bar{i}_+, \\ \frac{-b_i}{2}, & i \in I_+ \setminus \bar{i}_+, \\ 0, & \text{otherwise.} \end{cases}$$

All three points x^1, x^2, x^3 are feasible since $f(x^1) = f(x^2) = f(x^3) = 0$. Their midpoints x^{12}, x^{23}, x^{31}, however, are infeasible because $f(x^{12}) = 1$ and $f(x^{23}) = f(x^{31}) = \frac{1}{2}$. Therefore, F is not a union of two convex sets. □

Corollary 2.1. *Given a quadratic constraint in the form of constraint (8). If the constraint is not redundant, $|I_-| > 0$, and at least one of the conditions (C1)–(C3) is not satisfied, then F is not a union of two convex sets.*

Proof. If condition (C2) is not satisfied then Theorem 2.2 shows F is not a union of two convex sets. If condition (C2) is satisfied and condition (C3) is not satisfied, then Theorem 2.3, shows F is not a union of two convex sets. If both conditions (C2) and (C3) are satisfied and condition (C1) is not satisfied, then Theorem 2.4 shows F is not a union of two convex sets. □

Corollary 2.2. *Given a quadratic constraint in the form of constraint (8). If the constraint is not redundant, $|I_-| > 0$, and at least one of the conditions (C1)–(C3) is not satisfied, then F is not a union of a finite number of convex sets.*

Proof. We show that if constraint (8) is not redundant and does not satisfy one or more conditions (C1)–(C3), then for any given $k > 0, k \in \mathbb{N}$, F is not a union of $2k$ convex sets. The proof is similar to those for Theorems 2.2–2.4. Instead of finding three feasible points x^1, x^2, x^3, we can find $2k + 1$ feasible points $x^1, \ldots, x^{2k+1}$ such that for each pair x^i, x^j, the midpoint $\frac{1}{2}(x^i + x^j)$ is not feasible. Thus, x^i, x^j cannot be in the same convex set, and F must be a union of more than $2k$ convex sets.

We illustrate the proof of this theorem only for when $|I_-| > 0$ and $|I_0| > 0$ (analogous to Theorem 2.2). The other cases can be proved in a similar manner. Let $\bar{i}_- \in I_-$ and $\bar{i}_0 \in I_0$. Consider the system

$$RQ^T x = \bar{y}.$$

Let $x^{2\theta}, \theta = 0, \ldots, k$ be the unique solution to the system with

$$\bar{y}_i = \begin{cases} \sqrt{\theta} + \frac{b_i}{2}, & i = \bar{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \bar{i}_-, \\ \frac{-b_i}{2}, & i \in I_+, \\ \frac{\theta - h}{b_i}, & i = \bar{i}_0 \\ 0, & \text{otherwise.} \end{cases}$$

Let $x^{2\theta-1}, \theta = 1, \ldots, k$ be the unique solution to the system with

$$\bar{y}_i = \begin{cases} -\sqrt{\theta} + \frac{b_i}{2}, & i = \bar{i}_-, \\ \frac{b_i}{2}, & i \in I_- \setminus \bar{i}_-, \\ \frac{-b_i}{2}, & i \in I_+, \\ \frac{\theta - h}{b_i}, & i = \bar{i}_0, \\ 0, & \text{otherwise.} \end{cases}$$

All these points are feasible since $f(x^i) = 0, i = 0, \ldots, 2k$. However, $f(\frac{1}{2}(x^i + x^j)) > 0, i, j = 0, \ldots, 2k$ with $i \neq j$. Therefore, F is not the union of $2k$ convex sets. $\square$

2.3 Comments

If extra information is available from other constraints in the optimization problem, then constraints for which conditions (C1)–(C3) are not satisfied may still be reformulated by using second-order cones. Consider, for example, the problem

$$\begin{aligned} \min\, & x_1 + x_2 + x_4, \\ \text{s.t. } & x_1^2 - x_2^2 + x_3 + 10 \leq 0, \\ & 6x_1^2 - x_3 + 4x_4^2 + 4 \leq 0, \\ & x_1 + x_2 + x_4 \geq 2, \\ & x_3 \geq 0. \end{aligned}$$

Even though the first constraint does not satisfy condition (C2), we can reformulate the problem as

$$\begin{aligned} \min\, & x_1 + x_2 + x_4, \\ \text{s.t. } & x_1^2 - x_2^2 + t_3^2 + 10 \leq 0, \\ & 6x_1^2 - t_3^2 + 4x_4^2 + 4 \leq 0, \\ & x_1 + x_2 + x_4 \geq 2, \end{aligned}$$

which is obtained by substituting $x_3 = t_3^2$. The reformulated problem satisfies all conditions.

Similarly, in the presence of linear equality constraints, even if condition (C1) is not satisfied, we may perform a substitution. For example, consider the system

$$\begin{aligned} 7x_1^2 - x_2^2 - x_3^2 + 10 \leq 0, \\ x_2 + x_3 = 2, \end{aligned}$$

where the first constraint does not satisfy condition (C1). Substituting $x_3 = 2 - x_2$, we obtain the system

$$\begin{aligned} 7x_1^2 - 2(x_2 - 1)^2 + 8 \leq 0, \\ x_2 + x_3 = 2, \end{aligned}$$

which satisfies all our conditions.

Automating some of these reformulations is possible, but guaranteeing that all possible second-order cone constraints are identified would be computationally challenging. The difficulty arises in choosing the correct subset of substitutions to make from a large number of possible substitutions so that the result is either a convex constraint or a constraint representable as the union of two second-order cones.

3 Factorable Constraints

In this section, we consider problems that can be reformulated with constraints of the form

$$-\prod_{i=1}^{k} y_i + t \leq 0,$$

where $t \geq 0$ is a given constant and $y_i, i = 1, \ldots, k$, are original or auxiliary variables of the problem. Conditions sufficient for reformulating this constraint depend on whether t is positive or zero. These two cases differ in the assumptions on functions associated with the auxiliary variables.

3.1 Positive Constant

Consider a constraint of the form

$$-\left(\prod_{i=1}^{p} a_i^T x + b_i\right)\left(\prod_{j=1}^{q} g_j(x)\right)\left(\prod_{k=1}^{2r} h_k(x)\right) + t \leq 0, \tag{9}$$

where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for $i = 1, \ldots, p$ with the following assumptions:

(A1) $t \in \mathbb{R}$ is a given constant with $t > 0$,

(A2) $g_j : \mathbb{R}^n \to \mathbb{R}$ is a nonnegative concave function for $j = 1, \ldots, q$, and

(A3) $h_k : \mathbb{R}^n \to \mathbb{R}$ is a nonpositive convex function for $k = 1, \ldots, 2r$.

A simple case is a polynomial, with all real roots, constrained to be greater than a positive constant. An example of a nonnegative concave function is the constant elasticity of production production function introduced by Arrow et al. [1961] to study capital-labor substitution,

$$(\theta x_0^\sigma + (1-\theta)x_1^\sigma)^{\frac{1}{\sigma}},$$

where $0 \leq \theta \leq 1$ and $\sigma < 1$ are are constants.

To represent constraint (9) using second-order cones, we begin by forming the augmented system

$$\begin{aligned} -\prod_{i=1}^{p+q+2r} y_i + t \leq 0,& \\ y_i = a_i^T x + b_i,\ i = 1, \ldots, p,& \\ 0 \ \leq\ y_{p+j} \leq g_j(x), \quad j = 1, \ldots, q,& \\ 0 \ \leq\ -y_{p+q+k} \leq -h_k(x), \quad k = 1, \ldots, 2r.& \end{aligned} \tag{10}$$

Theorem 3.1. *Let assumptions (A1)–(A3) be satisfied. Then $x^* \in \mathbb{R}^n$ satisfies constraint (9) if and only if there exists $y^* \in \mathbb{R}^{p+q+2r}$ such that (x^*, y^*) satisfies the augmented system (10).*

Proof. Assume x^* satisfies constraint (9). Let $y_i^* = a_i^T x^* + b_i, i = 1, \ldots, p$, $y_{p+j}^* = g_j(x^*), j = 1, \ldots, q$, and $y_{p+q+k}^* = h_k(x^*), k = 1, \ldots, 2r$. Then (x^*, y^*) is feasible for the augmented system (10).

For the converse, let (x^*, y^*) be feasible for the augmented system (10). Assume $\prod_{i=1}^p y_i^* \leq 0$ or $y_{p+j}^* = 0$ for some $j = 1, \ldots, q$ or $y_{p+q+k}^* = 0$ for some $k = 1, \ldots, 2r$. In this case, $\prod_{i=1}^{p+q+2r}$ is nonpositive, contradicting our assumption of feasibility for the constraints (10).

Therefore, $\prod_{i=1}^p y_i^* > 0$, $y_{p+j}^* > 0, j = 1, \ldots, q$, and $y_{p+q+k}^* > 0, k = 1, \ldots, 2r$. Then,

$$\begin{aligned} t &\leq \prod_{i=1}^{p+q+2r} y_i^* \\ &= \left(\prod_{i=1}^{p} y_i^*\right) \times \left(\prod_{j=1}^{q} y_{p+j}^*\right) \times \left(\prod_{k=1}^{r}(-y_{p+q+2k-1}^*)(-y_{p+q+2k}^*)\right). \end{aligned}$$

Since $\prod_{i=1}^p y_i^* > 0$ and the other two products are positive, we have

$$\begin{aligned} t &\leq \left(\prod_{i=1}^{p} y_i^*\right) \times \left(\prod_{j=1}^{q} y_{p+j}^*\right) \times \left(\prod_{k=1}^{r}(-y_{p+q+2k-1}^*)(-y_{p+q+2k}^*)\right) \\ &\leq \left(\prod_{i=1}^{p} y_i^*\right) \times \left(\prod_{j=1}^{q} g_j(x^*)\right) \times \left(\prod_{k=1}^{r}(-h_{2k-1}(x^*))(-h_{2k}(x^*))\right) \\ &= \left(\prod_{i=1}^{p} a_i^T x^* + b_i\right) \times \left(\prod_{j=1}^{q} g_j(x^*)\right) \times \left(\prod_{k=1}^{2r} h_k(x^*)\right). \end{aligned}$$

Therefore, x^* satisfies constraint (9). □

All constraints in the augmented system (10) except the first, $t \leq \prod_{i=1}^{p+q+2r} y_i$, define a convex feasible region. We reformulate this first constraint using second-order cones. If it can be shown $y_i \geq 0, i = 1, \ldots, p$, then we can apply a reformulation technique similar to that used by Alizadeh and Goldfarb [2003] for maximizing the geometric mean of nonnegative affine functions to compute a second-order cone representation. We first describe this method and then extend it to the case when the variables $y_i, i = 1, \ldots p$, are unrestricted in sign.

Suppose $p+q+2r$ is even. If not, then introduce another variable $y_{p+q+2r+1}$, and modify the augmented system (10) to

$$\begin{aligned} t &\leq \prod_{i=1}^{p+q+2r+1} y_i, \\ y_i &= a_i^T x + b_i, \quad && i = 1,2,\ldots,p, \\ 0 \ \leq \ y_{p+j} &\leq g_j(x), && j = 1,\ldots,q, \\ 0 \ \leq \ -y_{p+q+k} &\leq -h_k(x), && k = 1,\ldots,2r, \\ y_{p+q+2r+1} &= 1. \end{aligned} \tag{11}$$

Under the assumption $y_i \geq 0, i = 1,\ldots,p$, we can transform system (11) by adding $\frac{p+q+2r}{2}$ new variables and constraints to obtain the system

$$\begin{aligned} \prod_{i=1}^{\frac{p+q+2r}{2}} w_i &\geq \sqrt{t}, \\ y_i &= a_i^T x + b_i, \\ 0 \ \leq \ y_{p+j} &\leq g_j(x), && j = 1,\ldots,q \\ 0 \ \leq \ -y_{p+q+k} &\leq -h_k(x), && k = 1,\ldots,2r \\ y_{2i-1}y_{2i} &\geq w_i^2, && i = 1,\ldots,\frac{p+q+2r}{2}, \\ w_i &\geq 0, && i = 1,\ldots,\frac{p+q+2r}{2}. \end{aligned} \tag{12}$$

The new constraints can then be rewritten as the second-order cones $y_{2i-1} + y_{2i} \geq \sqrt{(y_{2i-1} - y_{2i})^2 + (2w_i)^2}, i = 1,\ldots,\frac{p+q+2r}{2}$. After rewriting them, we obtain a system of inequalities similar to system (11), but with $\frac{p+q+2r}{2}$ variables in the product. We repeat this process $\log(p+q+2r)$ times, adding $\mathcal{O}(p+q+2r)$ new variables and second-order cone constraints in all to obtain a convex reformulation.

In general, the nonnegativity assumption $y_i \geq 0$, $i = 1,\ldots,p$, may not hold, and the above transformation may not be applicable to the original constraint (9). We now describe a branching procedure that can be applied to overcome this difficulty. Recall from Theorem 3.1 that for any feasible point of the augmented system (10), we have $y_i \neq 0, i = 1,\ldots,p$, $y_{p_j} > 0, j = 1,\ldots,q$, and $y_{p+q+k} > 0, k = 1,\ldots,2r$. Therefore, we create 2^p branches, each of which has either $y_i < 0$ or $y_i > 0$ for each $i = 1,\ldots,p$. The first branch has bounds $y_i < 0, i = 1,\ldots,p$, the second branch has bounds $y_1 < 0, y_i > 0, i = 2,\ldots,p$, and so on. The $(2^p)^{\text{th}}$ branch has the bounds $y_i > 0, i = 1,\ldots,p$. In all subproblems, $y_i > 0, i = p+1,\ldots,p+q+2r$, because their signs are fixed. We now consider any branch, and let $V_- = \{i \mid y_i < 0, i = 1,\ldots,p\}$ and $V_+ = \{i \mid y_i > 0, i = 1,\ldots,p+q+2r\}$. If $|V_-|$ is odd, then we prune the branch because it contains no feasible points. Thus, we prune half the branches: those with an odd number of negative y variables. Now consider any branch where $|V_-|$ is even. We make $\frac{|V_-|}{2}$ disjoint pairs from this set with an arbitrary ordering. For each pair (i,j), we introduce a new variable $w_{ij} \geq 0$ and constraint $y_i y_j \geq w_{ij}^2$. We make similar pairs for V_+, adding a new variable $w_{ij} \geq 0$ and constraint $y_i y_j \geq w_{ij}^2$ for each. We then add a constraint $\prod_{i,j} w_{ij} \geq \sqrt{t}$. The problem is now in the same form as system (12). The strict inequalities $w_{ij} > 0$ and $y_i < 0$ or $y_i > 0$ are not necessary because the constraint $\prod_{i,j} w_{ij} \geq \sqrt{t}$ automatically enforces them.

We illustrate the procedure using the following example:

$$\begin{aligned} &\min x_1^2 \\ s.t. \quad & x_1^3 x_2 - x_1 x_2^3 - x_1^2 x_2 + x_1 x_2^2 \geq 50. \end{aligned}$$

After observing that $x_1^3 x_2 - x_1 x_2^3 - x_1^2 x_2 + x_1 x_2^2 = x_1 x_2 (x_1 + x_2 - 1)(x_1 - x_2)$, we reformulate the problem as

$$\begin{aligned} \min \ & x_1^2 \\ s.t. \quad & y_1 y_2 y_3 y_4 \geq 50, \\ & y_1 = x_1, \\ & y_2 = x_2, \\ & y_3 = x_1 + x_2 - 1, \\ & y_4 = x_1 - x_2. \end{aligned}$$

We create $2^4 = 16$ branches as shown in Figure 1. In half the nodes, the number of y_i variables that are constrained to be negative is odd. These nodes $(2,3,5,8,9,12,14,15)$ can be pruned because the product $y_1 y_2 y_3 y_4$ is negative

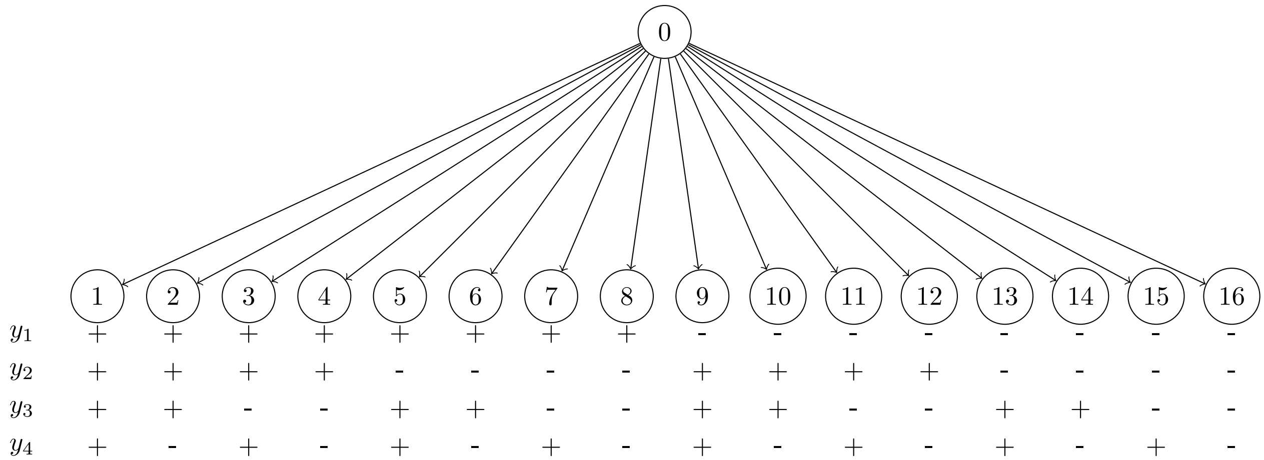

Figure 1: Search tree for the example described in Section 3. The $\pm$ sign below each node depicts the bound constraints on y_i. Nodes 2,3,5,8,9,12,14,15 are pruned immediately because $\prod_i y_i$ is nonpositive there.

there. For each of the remaining eight nodes, we reformulate the subproblem using second-order cone constraints. Consider, for example, node 6. An equivalent reformulation there is

$$
\begin{aligned}
&\min\ x_1^2 \\
s.t.\quad & w_1 w_2 \geq \sqrt{50}, \\
& y_1 = x_1, \\
& y_2 = x_2, \\
& y_3 = x_1 + x_2 - 1, \\
& y_4 = x_1 - x_2, \\
& y_1 + y_3 \geq \sqrt{(y_1 - y_3)^2 + (2w_1)^2} \qquad (\iff y_1 y_3 \geq w_1^2), \\
& -(y_2 + y_4) \geq \sqrt{(y_2 - y_4)^2 + (2w_2)^2} \qquad (\iff y_2 y_4 \geq w_2^2), \\
& w_1, w_2 \geq 0, \\
& y_1, y_3 \geq 0, \\
& y_2, y_4 \leq 0.
\end{aligned}
$$

The first constraint now has a product of two variables and is reformulated as $w_1 + w_2 \geq \sqrt{(w_1 - w_2)^2 + 4\sqrt{50}}$. The resulting optimization problem is now convex.

To apply this method to a general constraint, we need to factor the polynomials and nonlinear expressions. General polynomials can be factored over finite rings, like the ring of integers or rational numbers, by using algorithms of Wang [1976] or Lenstra [1984]. Such algorithms have been implemented in open-source software such as Singular [Decker et al., 2010] and Maxima [Maxima, 2009]. However, these software packages may be unable to find linear factors of a polynomial either because it has irrational roots (e.g., $x_1^2 - 2$) or it does not have real roots (e.g., $x_1^4 x_2 + x_2^5 + 2x_2$). Our method also does not work in such cases, and we must resort to more general-purpose global optimization methods. Our technique also does not work when the constant t is nonpositive.

3.2 Zero Constant

We now consider the case when the constant t of constraint (9) is zero. In this case Theorem 3.1 is not applicable. In particular, if $t = 0$, we can have a feasible point (x^*, y^*) for the augmented system (10) with $\prod_{i=1}^p y_i^* < 0$, $\prod_{j=1}^q y_j^* = 0$, and $\prod_{j=1}^q g_j(x^*) y_j^* = 0$, which is not feasible for the original constraints (10). To address this problem, we would need to enforce $y_{p+j} = g_j(x), j = 1, \ldots, q$ and $y_{p+q+k} = h_k(x), k = 1, \ldots, 2r$, leading to a nonconvex formulation unless $g_j(x)$ and $h_k(x)$ are linear functions.

Therefore, to obtain convex regions after branching, we consider the constraint

$$
-\left(\prod_{i=1}^{p} a_i^T x + b_i\right)\left(\prod_{j=1}^{q} g_j(x)\right)\left(\prod_{k=1}^{2r} h_k(x)\right) \leq 0, \tag{13}
$$

where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for $i = 1, \ldots, p$ with the following assumptions:

(B1) $g_j : \mathbb{R}^n \to \mathbb{R}$ is a strictly positive function for $j = 1, \ldots, q$, and

(B2) $h_k : \mathbb{R}^n \to \mathbb{R}$ is a strictly negative function for $k = 1, \ldots, 2r$.

In this case, we do not need any assumptions about the convexity of the functions g_j and h_k; the constraint is equivalent to the system

$$-\prod_{i=1}^{p} y_i \leq 0,$$
$$y_i = a_i^T x + b_i, \quad i = 1, \ldots, p.$$

We do not need any second-order cone reformulations in this case because the feasible region associated with the first constraint is a union of orthants, each of which can be expressed by using simple bounds on the y variables. We first handle the case when one or more y variables are zero, creating p branches in which one of the y_i variables is constrained to zero and the remaining variables are unrestricted in sign. The j^{th} branch has the linear formulation

$$\begin{aligned} & y_i = a_i^T x + b_i, \, i = 1, \ldots, p, \\ & y_j = 0. \end{aligned}$$

As in Section 3.1, we then create 2^p additional branches: $y_i \leq 0$ for all i in the first branch, $y_1 \leq 0, y_i \geq 0, i = 1, \ldots, p$ in the next, and so on. The l^{th} branch has only the following linear constraints,

$$\begin{aligned} & y_i = a_i^T x + b_i, \, i = 1, \ldots, p, \\ & y_i \leq 0, \qquad i \in V_-^l, \\ & y_i \geq 0, \qquad i \in V_+^l, \end{aligned}$$

where V_-^l, V_+^l are the index sets of variables that are constrained to be nonpositive or nonnegative in branch l, respectively. We prune those branches in which $|V_-^l|$ is odd because at least one of the variables must be zero to obtain feasibility for the original problem, and these cases are covered by the first set of branches. The branching used is sufficient to solve these instances even though the branches do not have mutually disjoint feasible regions.

To illustrate our procedure, we consider a modification of the example in the previous section:

$$\begin{aligned} & \min x_1^2 \\ s.t. \quad & x_1^3 x_2 - x_1 x_2^3 - x_1^2 x_2 + x_1 x_2^2 \geq 0. \end{aligned}$$

As before, we reformulate the problem as

$$\begin{aligned} \min \ & x_1^2 \\ s.t. \quad & y_1 y_2 y_3 y_4 \geq 0, \\ & y_1 = x_1, \\ & y_2 = x_2, \\ & y_3 = x_1 + x_2 - 1, \\ & y_4 = x_1 - x_2. \end{aligned}$$

We again create $2^4 = 16$ branches as shown in Figure 1. The formulation in Node 6, for example, is

$$\begin{aligned} \min \ & x_1^2 \\ & y_1 = x_1, \\ & y_2 = x_2, \\ & y_3 = x_1 + x_2 - 1, \\ & y_4 = x_1 - x_2, \\ & y_1, y_3 \geq 0, \\ & y_2, y_4 \leq 0. \end{aligned}$$

In addition, we have four more branches, where the i^{th} new branch $(i = 1, \ldots, 4)$ has the formulation

$$\begin{aligned} \min \ & x_1^2 \\ & y_1 = x_1, \\ & y_2 = x_2, \\ & y_3 = x_1 + x_2 - 1, \\ & y_4 = x_1 - x_2, \\ & y_i = 0. \end{aligned}$$

4 Computational Experiments

To test the usefulness of exploiting second-order cone structure for global optimization, we created several small instances of three types:

1. Instances that have only quadratic and linear functions in the constraints. The quadratic constraints meet the three assumptions described in Section 2. Their names follow the convention "qXdY", where "X" denotes the number of quadratic constraints, and "Y" denotes the total number of variables.
2. Instances that have constraints with quadratic functions plus exponential terms like those in Example 1 or univariate variables with powers at least two like those in Example 2 (Section 2). Instance of the first type are named "pXdYe", with "X" being the number of constraints and "Y" being the total number of variables. The second type are named "pXdY". All the "pXdY" instances have variables with degree four in some constraints.
3. Instances that have constraints with factorable polynomials and $t > 0$ as described in Section 3.1. These are named "fXdY", with "X" being the number of constraints, and "Y" being the total number of variables. We add a suffix "b" to distinguish two instances having the same dimensions, but different factors. We restrict to the special case when we do not have any nonlinear factors, i.e., $q = r = 0$, because we do not know of any general methods for factoring them.

These instances are available online at the NEOS wiki[1] in AMPL (.mod) and GAMS (.gms) formats.

We compared the performance of our techniques against two well-known global optimization software packages: BARON, version 9.0.5 [Sahinidis, 1996], and Couenne, version 0.3.2 [Belotti, 2009]. BARON used CPLEX [IBM, 2009] as the linear programming solver and MINOS [Murtagh and Saunders, 1998] as the nonlinear programming solver. Couenne was compiled with CLP [Forrest, 2010] as the linear programming solver and IPOPT [Wächter and Biegler, 2006] as the nonlinear programming solver. All experiments were performed on a computer with a 2.66 GHz Intel Xeon CPU having 4 MB cache and Ubuntu 8.04.4 operating system. We used two stopping criteria: gap and time. We stop solving when the gap percentage falls to 0.01%, that is, when $\left|\frac{ub-lb}{ub}\right| \leq 0.0001$, where lb is the lower bound and ub is the upper bound. We also impose a time limit of one hour.

We implemented our routines so that they can be easily incorporated into other global optimization solvers. We first read the instance provided in the AMPL format and traverse the computational graph [Gay, 2005] of its representation to recognize the structure. If a constraint has only a quadratic and linear functions, then we use LAPACK [Anderson et al., 1999] to compute the eigenvalues and eigenvectors of the Hessian of the quadratic and check for the three conditions (C1)–(C3) described in Section 2. Similarly, if we recognize a general polynomial in a constraint, we try to factor it using Singular. If the polynomial can be factored into linear factors, we proceed with the techniques described in Section 3. For the above two classes of problems, our implementation works automatically, without any manual intervention.

When in addition to the quadratic terms, we have terms that are either exponential ("pXdYe") or higher-order univariate terms ("pXdY"), we first find the eigenvalues and eigenvectors associated with the quadratic as before. Then, we reformulate the instance as a convex nonlinear program with binary variables, similar to system (7), but with additional convex constraints: $\alpha^{\frac{1}{2}r_i^T x} - x_{n+i} \leq 0$ for exponential functions (Example 1) and $\left|r_i^T x\right|^{\frac{p_i}{2}} \leq x_{n+i}$ for higher-order polynomials (Example 2). We solve the reformulations using a simple branch-and-bound procedure. We need branch only on the binary variables because the remaining problem is convex. The first- and second-order derivatives are available directly from AMPL. Since routines for traversing the computational graphs are already available in global optimization solvers such as BARON and Couenne, it would be easy for the developers to incorporate our procedure in those solvers.

We use IPOPT for solving the convex nonlinear programs obtained after branching. Since IPOPT requires first- and second-order derivatives, we implemented functions to calculate these values at a given point. The only function that we need to implement is that describing a second-order cone,

$$f(y_0, y) = \sqrt{\left(\sum_{i=1}^{n} y_i^2 + K\right)} - y_0.$$

This function is twice continuously differentiable everywhere except when $K = 0$ and $y = 0$. At this exceptional point, the gradient in the limit $K \to 0$ is 0. The elements of Hessian at this point become unbounded in the limit. Hence, if $K = 0$ and if the nonlinear solver asks for evaluating derivatives at 0, we do not proceed further and return failure. For the instances of our test set, such an event never occurs. In any case, if gradients are available, a robust nonlinear optimization solver will converge if we supply a Hessian approximation in the case when $y = 0$ and $K = 0$.

We list the results with BARON, Couenne, and our methods in Table 1. We observe that while both BARON and Couenne fail to solve several instances in the time limit, we solve all problems in a few seconds. The improvement grows as the number of nonlinear constraints in the instance increases. For some instances, the gap

[1] http://wiki.mcs.anl.gov/NEOS/index.php/LIGOLIB

Table 1: Size of the instances and performance of BARON and Couenne as compared with our implementation (q-soc). The number following the ">" under "# Nodes" denotes the number of nodes created when the time limit of one hour was reached. Gap% is defined as $\left|\frac{ub-lb}{ub}\right| \times 100$.

			# Nodes			Total time or gap% at 1h			Gap% at 60s	
Inst.	Vars	Cons	BARON	Couenne	q-soc	BARON	Couenne	q-soc	BARON	Couenne
q1d2	2	2	39	14	2	0.04s	0.2s	0.02s	0.0	0.0
q1d3	3	2	297	701	2	0.4s	0.7s	0.03s	0.0	0.0
q2d6	6	4	697559	>5081900	4	841.2s	0.1%	0.2s	7.2	11.8
q3d6	6	6	339	1801	8	0.6s	1.3s	0.7s	0.0	0.0
q3d9	9	6	>1250000	>2921300	8	17.6%	21.1%	0.3s	607.1	1770.6
q4d8	8	8	33845	26901	16	46.2s	15.4s	2.4s	0.0	0.0
q5d10	10	10	1280974	2432401	32	1718.3s	1561.5s	1.8s	0.9	3.5
q5d10b	10	10	>1033800	>3182100	32	123.3%	56.4%	1.8s	930.3	662.1
q5d15	15	10	256889	>1398400	32	1113.8s	23.1%	18.4s	213.3	577.8
q6d12	12	12	>1410000	>4000900	64	0.5%	3.5%	9.0s	8.1	4.1
p4d12	12	12	1251	11501	31	8.6s	23.1s	0.8s	0.0	0.0
p5d10	10	10	>996000	>2299500	11	71.2%	27.6%	2.1s	595.7	769.3
p5d10e	10	10	>1031000	>294600	11	73.5%	290.8%	1.2s	454.4	760.9
p5d15	15	10	6269	93601	63	62.5s	213.9s	1.4s	0.1	52.4
p5d15e	15	10	283915	>213600	63	2751.0s	9.1%	1.3s	291.7	140.6
p6d18	18	12	212599	>979200	91	2445.7s	273.8%	2.7s	235.3	575.0
f1d3	3	2	1497	66801	4	2.4s	47.4s	0.2s	0.0	0.0
f2d4	4	4	2349	5901	64	5.3s	9.1s	1.9s	0.0	0.0
f2d6	6	4	447363	>1933300	16	1417.3s	1.2%	0.6s	5.5	19.1
f3d6	6	6	81167	801001	512	266.0s	1572.5s	20.1s	416.6	497.8
f3d6b	6	6	915	9001	512	1.5s	8.6s	22.4s	0.0	0.0
f3d9	9	6	>550000	>1208700	64	4.9%	19.8%	3.1s	29.4	33.3
f4d8	8	8	>687000	>1636700	4096	293.3%	392.8%	195.8s	902.3	826.3

was more than 100% because the best lower bound obtained by the solvers was negative while the best upper bound was positive. We found that for these small instances, while both BARON and Couenne could find the best possible upper bound quickly, it was difficult for them to increase the lower bound. The several orders of magnitude improvement we observe in these instances provides evidence that such reformulation techniques can be useful in solving such problems to global optimality.

5 Conclusions

In this paper we presented techniques for automatically reformulating quadratic constraints that satisfy certain assumptions in such a way that we can obtain convex second-order cone formulations after branching. Our preliminary experiments on small instances show that when such techniques are not used, a global optimization solver can perform poorly even on small instances. Such techniques can be incorporated in current solvers, and similar techniques can be developed for more classes of constraints. Effectively implementing these methods poses several challenges, however. If other nonconvex constraints appear in the instance, for example, then solving nonlinear programming relaxations may not be as fast as solving linear programming relaxations.

Such techniques can be extended to other nonlinear functions. We have focused on reformulating nonconvex programs using second-order cones. We can also reformulate constraints by using generalized cones

$$\|\tilde{A}x + \tilde{b}\|_p \leq |\tilde{c}^T x + \tilde{d}|,$$

where $p \geq 1$. The feasible region for the above constraint is a union of two p^{th} order cones and could be used to represent higher-order polynomials. In Example 2, for example, we assumed that the higher-order terms exist only as a sum of univariate monomials. This idea can be extended to general polynomials if we can reformulate a constraint of the form

$$S(x) \leq 0,$$

where $S(x) : \mathbb{R}^n \to \mathbb{R}$ is a polynomial of degree $p \in \mathbb{N}$, as

$$\sum_{i=1}^{k} |y_i + b_i|^p \leq |y_1 + b_1|^p .$$

Unfortunately, when $p > 2$, general methods for converting a polynomial $S(x)$ into the above form are not known. Progress has been made in this direction, however, for the case when $p = 3$. See, for example, the work of Kolda and Bader [2007].

We analyze the quadratic and nonlinear functions only at the root node when checking if we can reformulate the constraints. We could check for the required structures as the branch-and-bound tree is generated. Since the nodes represent a restriction of the original problem, we may be able to reformulate the problem at one of those nodes even if the problem violates the necessary conditions at the root node.

Acknowledgments

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357. This work was also supported by the U.S. Department of Energy through grant DE-FG02-05ER25694.

References

F. Al-Khayyal, C. Larsen, and T. Voorhis. A Relaxation Method for Nonconvex Quadratically Constrained Quadratic Programs. *Journal of Global Optimization*, 6:215–230, 1995. doi: 10.1007/BF01099462.

F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, 95(3):3–51, 2003. doi: 10.1007/s10107-002-0339-5.

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. ISBN 0-89871-447-8 (paperback).

I. P. Androulakis, C. D. Maranas, and C. A. Floudas. αBB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, 7(4):337–363, 1995. doi: 10.1007/BF01099647.

K. J. Arrow, H. B. Chenery, B. S. Minhas, and R. M. Solow. Capital-labor substitution and economic efficiency. *The Review of Economics and Statistics*, 43(3):225–250, 1961. doi: 10.2307/1927286.

P. Belotti. *Couenne: A User's Manual*, 2009. Available online at `https://projects.coin-or.org/Couenne/`.

P. Belotti. Disjunctive Cuts for Non-Convex MINLP. In *IMA Volume Series*, 2010. Accepted.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

W. Decker, G. M. Greuel, G. Pfister, and H. Schönemann. Singular 3-1-1 — A computer algebra system for polynomial computations. 2010. Available online at `http://www.singular.uni-kl.de`.

J. Forrest. *COIN-OR Linear Programming solver*, 2010. Available online at `https://projects.coin-or.org/Clp`.

D. M. Gay. Writing **.nl** files. Technical Report 2005-7907P, Sandia National Laboratories, Albuquerque, NM, 2005.

IBM. *User's manual for CPLEX V12.1*, 2009. Available online at `http://ibm.com/software/integration/optimization/cplex/`.

T. Kolda and B. Bader. Tensor decompositions and applications. Technical Report SAND2007-6702, Sandia National Laboratories, Albuquerque, NM, Nov. 2007.

A. K. Lenstra. Factoring multivariate polynomials over algebraic number fields. In M. P. Chytil and V. Koubek, editors, *Mathematical Foundations of Computer Science*, pages 389–396, 1984. doi: 10.1007/BFb0030321.

Maxima. *A Computer Algebra System. Version 5.18.1*, 2009. Available online at `http://maxima.sourceforge.net`.

B. A. Murtagh and M. A. Saunders. MINOS 5.4 user's guide. Report SOL 83-20R, Department of Operations Research, Stanford University, 1998.

N. V. Sahinidis. BARON: A general purpose global optimization software. *Journal of Global Optimization*, 8:201–205, 1996. doi: 10.1007/BF00138693.

A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: Projected formulations. Technical Report RC24695, IBM Research Division, 2008.

M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004. doi: 10.1007/s10107-003-0467-6.

R. C. Thompson. The behavior of eigenvalues and singular values under perturbations of restricted rank. *Linear Algebra and Its Applications*, 13:69–78, 1976. doi: 10.1007/BF01099647.

A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006. doi: 10.1007/s10107-004-0559-y.

P. Wang. Factoring multivariate polynomials over algebraic number fields. *Mathematics of Computation*, 30(134): 324–336, 1976. doi: 10.1145/1088309.1088316.

J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1965.