Mathematical Programming
ISyE Department, University of Wisconsin-Madison

Oktay Günlük[1] · Jeff Linderoth[2]

Perspective Reformulations of Mixed Integer Nonlinear Programs with Indicator Variables

February 10, 2009

Abstract. We study mixed integer nonlinear programs (MINLP)s that are driven by a collection of indicator variables where each indicator variable controls a subset of the decision variables. An indicator variable, when it is "turned off", forces some of the decision variables to assume fixed values, and, when it is "turned on", forces them to belong to a convex set. Many practical MINLPs contain integer variables of this type. We first study a mixed integer set defined by a single separable quadratic constraint and a collection of variable upper and lower bound constraints, and a convex hull description of this set is derived. We then extend this result to produce an explicit characterization of the convex hull of the union of a point and a bounded convex set defined by analytic functions. Further, we show that for many classes of problems, the convex hull can be expressed via conic quadratic constraints, and thus relaxations can be solved via second-order cone programming. Our work is closely related with the earlier work of Ceria and Soares (1999) as well as recent work by Frangioni and Gentile (2006) and, Aktürk, Atamtürk and Gürel (2007). Finally, we apply our results to develop tight formulations of mixed integer nonlinear programs in which the nonlinear functions are separable and convex and in which indicator variables play an important role. In particular, we present computational results for three applications – quadratic facility location, network design with congestion, and portfolio optimization with buy-in thresholds – that show the power of the reformulation technique.

Key words. Mixed-integer nonlinear programming – perspective functions

1. Introduction

A popular and effective approach to solving mixed integer nonlinear programs (MINLP)s is to approximate the continuous relaxation of the MINLP with some form of linearization and to use this relaxation in an enumeration algorithm [30, 9, 1]. Since software for nonlinear programs continues to become more efficient and robust, it is natural to consider using strong non-linear relaxations of the MINLP in algorithms instead. In this paper, we describe a simple and fairly general scheme to strengthen non-linear relaxations of a class of {0,1}-mixed integer nonlinear programs. Our approach is complementary to linear and non-linear cutting approaches as it can be used together with cuts.

Mathematical Sciences Department, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, USA, e-mail: `gunluk@us.ibm.com`

Department of Industrial and Systems Engineering,University of Wisconsin-Madison, 1513 University Avenue, Madison, WI 53706, USA, e-mail: `linderoth@wisc.edu`

1.1. Perspective Reformulation

We study MINLPs that are driven by a collection of indicator variables where each indicator variable controls a subset of the decision variables. In particular, we are interested in MINLPs where an indicator variable, when it is "turned off", forces some of the decision variables to assume fixed values, and, when it is "turned on", forces them to belong to a convex set. We call such programs *indicator-induced {0,1}-mixed integer nonlinear programs.*

A generic indicator-induced {0-1}-MINLP can be written as

$$z^* \stackrel{\text{def}}{=} \min_{(x,z)\in X\times(Z\cap\mathbb{B}^{|I|})} \{c^T x + d^T z \mid g_j(x,z) \le 0 \ \forall j \in M,\ (x_{V_i}, z_i) \in S_i \ \forall i \in I\}, \quad (1)$$

where z are the indicator variables, x are the continuous variables and x_{V_i} denotes the collection of continuous variables (i.e. $x_j,\ j \in V_i$) controlled by the indicator variable z_i. In the formulation, the sets may intersect, that is, for some $i \neq j$ we can have $V_i \cap V_j \neq \emptyset$. Sets $X \subseteq \mathbb{R}^n$ and $Z \subseteq \mathbb{R}^{|I|}$ are polyhedral sets of appropriate dimension and S_i is the set of points that satisfy all constraints associated with the indicator variable z_i:

$$S_i \stackrel{\text{def}}{=} \left\{ (x_{V_i}, z_i) \in \mathbb{R}^{|V_i|} \times \mathbb{B} \ \middle| \ \begin{array}{ll} x_{V_i} = \hat{x}_{V_i} & \text{if } z_i = 0 \\ x_{V_i} \in \Gamma_i & \text{if } z_i = 1 \end{array} \right\},$$

where

$$\Gamma_i \stackrel{\text{def}}{=} \{x_{V_i} \in \mathbb{R}^{|V_i|} \mid f_j(x_{V_i}) \le 0 \ \forall j \in C_i,\ u_k \ge x_k \ge \ell_k \ \forall k \in V_i\}$$

is bounded for all $i \in I$. Notice that, due to the definition of S_i, we have $z_i \in \{0,1\}$ for all $i \in I$. The objective function in (1) is assumed to be linear without loss of generality. If necessary, an additional variable can be used to move the nonlinearity from the objective function to the constraint set.

In this paper we study the convex hull description of the sets S_i when Γ_i is a convex set. An important observation is that Γ_i can be a convex set even when some of the functions f_j defining the set are non-convex. Let $S_i^c = \text{conv}(S_i)$. Using S_i^c, one can write a "tight" continuous relaxation of (1) as

$$z^{PR} \stackrel{\text{def}}{=} \min_{(x,z)\in X\times Z} \{c^T x + d^T z \mid g_j(x,z) \le 0 \ \forall j \in M,\ (x_{V_i}, z_i) \in S_i^c \ \forall i \in I\}, \quad (2)$$

where S_i in (1) is replaced by its convex hull. We call (2) the *perspective relaxation* of (1), as the description of S_i^c involves *perspective* functions, as described subsequently in Section 3.

When all f_j are convex and bounded for $j \in C_i$, another convex relaxation of S_i can simply be obtained as follows:

$$\begin{aligned} S_i^R \stackrel{\text{def}}{=} \{x_{V_i} \in \mathbb{R}^{|V_i|} \mid\ & f_j(x_{V_i}) \le (1 - z_i) f_j(\hat{x}_{V_i}) \ \forall j \in C_i, \\ & u_k z_i \ge x_k - (1 - z_k)\hat{x}_{V_i} \ge \ell_k z_i \ \forall k \in V_i\}, \end{aligned}$$

which leads to what we call the *natural* continuous relaxation of (1):

$$z^{NR} \stackrel{\text{def}}{=} \min_{(x,z)\in X\times Z} \{c^T x + d^T z \mid g_j(x,z) \le 0 \ \forall j \in M,\ (x_{V_i}, z_i) \in S_i^R \ \forall i \in I\} \quad (3)$$

where S_i in (1) is replaced with S_i^R. Notice that as S_i^R is convex and $S_i \subset S_i^R$, we have $S_i^c \subseteq S_i^R$ for all $i \in I$. Therefore,

$$z^* \ge z^{PR} \ge z^{NR}.$$

In general, as S_i^c is the smallest convex set that contains S_i, the perspective relaxation (2) leads to an effective computational approach provided that (i) it can be solved efficiently, and, (ii) it gives a good approximation of z^*. We later present computational results that show that this indeed is the case for a number of problems. We also show that in some cases, S_i^c is representable as a quadratic cone and this improves computational effectiveness of our approach even further.

1.2. Literature Review

There has been some recent work on generating strong relaxations for convex MINLPs. One line of work has been on extending general classes of cutting planes from mixed integer linear programs. Specifically, Stubbs and Mehrotra [31] explain how the disjunctive cutting planes of Balas et al. [4] can be applied for MINLP, Cezik and Iyengar [13] extend the Gomory cuts [16], and Atamtürk and Narayanan [3] extend the mixed integer rounding cuts of Nemhauser and Wolsey [28] to conic mixed integer programs. A second line of work has focused on generating problem specific cutting planes, for example see Günlük et al. [20]. In some cases these inequalities can be used to strengthen the perspective relaxation even further.

Related to this work, Frangioni and Gentile [14] have introduced a class of linear inequalities called *perspective cuts* for a class of indicator-induced MINLPs. As we discuss in Section 4.2, perspective cuts are outer approximation cuts for S_i^c and therefore the perspective relaxation (2) can be viewed as implicitly including all (infinitely many) perspective cuts to a straightforward relaxation of (1). Another related work is that of Grossmann and Lee [17], who extend the convex hull characterization of Ceria and Soares [12] to general (convex) disjunctive programs. The characterization relies on perspective functions. Concurrent with this work, Aktürk et al. [2] independently gave a strong characterization of S_i^c when $\Gamma_i = \{x \in \mathbb{R}^2 \mid x_1^t - x_2 \le 0,\ u \ge x_1, x_2 \ge 0\}$ for $t \ge 1$. They use this characterization in an algorithm to solve nonlinear machine scheduling problems.

1.3. Motivation and Contribution

A main purpose of this work is to demonstrate the application of concepts successfully used in mixed-integer linear programming (MILP) to MINLP. For example, successful commercial software for MILP recognizes structure and uses problem reformulation and cutting planes to build tight continuous relaxations. To apply this idea in MINLP, we analyze simple sets that form substructures in many practical MINLPs. Based on our analysis we propose a reformulation method to produce extended formulations for these sets that yield strong relaxations. Our overarching goal is to demonstrate the power of these techniques and encourage software developers for MINLP to include automatic reformulation techniques in their solvers. Even though many of the ingredients we use in our reformulation can be found in the literature, this has not yet translated into making MINLP solvers more effective. In Section 5 we demonstrate that commercially available MINLP solvers fail to solve certain problems unless the reformulation ideas we discuss are incorporated.

The remainder of the paper is divided in five sections. In Section 2, we study a mixed integer set defined by a single separable quadratic constraint and a collection of variable upper and lower bound constraints. In Section 3, we extend our observations to more general sets. Section 4 discusses connections between our work and earlier work by Ceria and Soares [12] and Frangioni and Gentile [14]. Finally in Section 5, we demonstrate the strength of our reformulation ideas by applying it to three problems: a quadratic uncapacitated facility location problem, a network design problem with nonlinear congestion constraints and a portfolio optimization model with buy-in thresholds. Some conclusions are offered in Section 6.

2. A Quadratic Set with Variable Bounds

The purpose of this section is to present a convex hull description of the set:

$$Q = \Big\{(w, x, z) \in \mathbb{R}^{n+1} \times \mathbb{B}^n : w \geq \sum_{i=1}^{n} q_i x_i^2,\ u_i z_i \geq x_i \geq l_i z_i,\ i \in I\Big\}, \tag{4}$$

where $I = \{1, \ldots, n\}$ and $q, u, l \in \mathbb{R}^n_+$.

To our knowledge, the first convex hull description of Q was stated without proof in the unpublished Ph.D. thesis of Stubbs [32]. The convex hull description of Q is closely related to the convex envelope of the function $\sum_{i=1}^n q_i x_i^2$ over a mixed integer set. Consequently, the results presented in this section could also be derived using global optimization terminology and literature. (See [21] for a good introduction to global optimization). However, we prefer to derive the results from first principles to demonstrate that "standard" techniques from the mixed integer linear programming (MILP) literature can also be applicable to MINLP. Building on intuition gained from our study of Q, we are able to derive the convex hull of more general mixed integer nonlinear sets in Section 3.

2.1. A Low Dimensional Analogue

To understand the set Q, we first study a simpler mixed-integer set with only 3 variables, which can be obtained by setting $n = 1$ and $q_1 = 1$ in (4). Let

$$S = \Big\{(x, y, z) \in \mathbb{R}^2 \times \mathbb{B} \,:\, y \geq x^2, \;\; uz \geq x \geq lz, \; x \geq 0\Big\},$$

where $u, l \in \mathbb{R}$. In Lemma 1 we show that the convex hull of S is given by

$$S^c \;=\; \big\{(x, y, z) \in \mathbb{R}^3 \,:\, yz \geq x^2, \; uz \geq x \geq lz, \; 1 \geq z \geq 0, \; x, y \geq 0\big\}\,.$$

Geometrically, the set S^c consists of all points that lie above a line segment connecting the origin to the point $(t, t^2, 1)$ for each $t \geq 0$. See Figure 1.

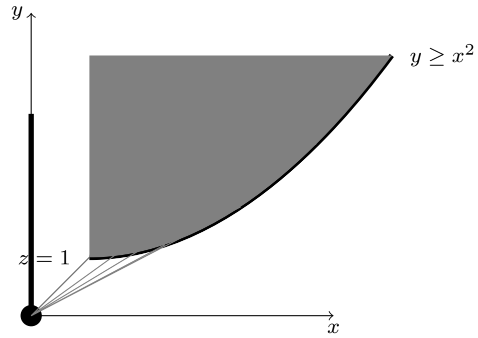

Fig. 1. The set S^c

Note that even though $x^2 - yz$ is not a convex function the set $T^c = \{(x, y, z) \in \mathbb{R}^3 : yz \geq x^2, \; x, y, z \geq 0\}$ is convex (see, Günlük and Linderoth [19]) and therefore S^c, obtained by intersecting T^c with half-spaces, is also convex.

Lemma 1. $\operatorname{conv}(S) \;=\; S^c$.

Proof. First note that $S = S^0 \cup S^1$ where $S^0 \;=\; \big\{(0, y, 0) \in \mathbb{R}^3 \,:\, y \geq 0\big\}$, and

$$S^1 = \;\big\{(x, y, 1) \in \mathbb{R}^3 \,:\, y \geq x^2, \; u \geq x \geq l, \; x \geq 0\big\}\,.$$

As $S^0, S^1 \subset S^c$ and S^c is a convex set, we have $\operatorname{conv}(S) \subseteq S^c$.

Next, consider a point $\bar{p} = (\bar{x}, \bar{y}, \bar{z}) \in S^c$. If $\bar{z} = 0$, then $\bar{p} = (0, \bar{y}, 0)$ where $\bar{y} \geq 0$ and $\bar{p} \in S^0$. If, on the other hand, $\bar{z} \neq 0$, then $\bar{p} = p' + d$ where $p' = (\bar{x}, \bar{x}^2/\bar{z}, \bar{z}) \in S^c$ and $d = (0, \bar{y} - \bar{x}^2/\bar{z}, 0) \geq 0$. Furthermore, $p' = (1 - \bar{z})p_0 + \bar{z}p_1$ where $p_0 = (0, 0, 0) \in S^0$ and $p_1 = (\bar{x}/\bar{z}, \bar{x}^2/\bar{z}^2, 1) \in S^1$. As $1 \geq \bar{z} \geq 0$, we have $p' \in \operatorname{conv}(S)$. In addition, as $(0, 1, 0)$ is an (extreme) direction of S^0 and S^1, it is a direction of $\operatorname{conv}(S)$, implying $\bar{p} \in \operatorname{conv}(S)$. Therefore $S^c \subseteq \operatorname{conv}(S)$. ∎

2.2. An Extended Formulation for Q

Consider the following extended formulation of Q

$$\bar{Q} \stackrel{\text{def}}{=} \Big\{(w,x,y,z) \in \mathbb{R}^{3n+1} \;:\; w \geq \sum_i q_i y_i, (x_i, y_i, z_i) \in S_i, \quad i = 1, 2, \ldots, n\Big\}$$

where S_i has the same form as the set S discussed in the previous section except the bounds u and l are replaced with u_i and l_i. Note that if $(w,x,y,z) \in \bar{Q}$ then $(w,x,z) \in Q$, and therefore $\text{proj}_{(w,x,z)}(\bar{Q}) \subseteq Q$. On the other hand, for any $(w,x,z) \in Q$, letting let $y_i' = x_i^2$ gives a point $(w,x,y',z) \in \bar{Q}$. Therefore, $\bar{Q}$ is indeed an extended formulation of Q, or, in other words, $Q = \text{proj}_{(w,x,z)}(\bar{Q})$.

Before we present a convex hull description of $\bar{Q}$ we first define some basic properties of mixed-integer sets. Using these definitions, we then show some elementary observations which are known for polyhedral sets.

Definition 1. *Let $P \subset \mathbb{R}^n$ be a closed set and let $p \in P$.*

(i) p is called an extreme point *of P if it can not be represented as $p = 1/2p_1 + 1/2p_2$ for $p_1, p_2 \in P$, $p_1 \neq p_2$. Set P is called* pointed *if it has extreme points.*
(ii) P is called integral *with respect to a subset of the indices $I \subseteq \{1, \ldots, n\}$ if for any extreme point $p \in P$, $p_i \in \mathbb{Z}$ for all $i \in I$.*

Lemma 2. *For $i = 1, 2$ let $P_i \subset \mathbb{R}^{n_i}$ be a closed and pointed set which is integral with respect to indices I_i. Let $P' = \{(x,y) \in \mathbb{R}^{n_1+n_2} \;:\; x \in P_1,\; y \in P_2\}$.*
(i) P' is integral with respect to $I_1 \cup I_2$.
(ii) $\text{conv}(P') = \{(x,y) \in \mathbb{R}^{n_1+n_2} \;:\; x \in \text{conv}(P_1),\; y \in \text{conv}(P_2)\}$.

Proof. (i) A point $p = (x', y')$ is an extreme point of P' if and only if x' is an extreme point of P_1 and y' is an extreme point of P_2. As all extreme points of P_1 and P_2 are integral, p is integral as well.

(ii) Similarly, $p = (x', y') \in conv(P)$ if and only if $(x', y') = \sum_j \lambda_j (x_j, y_j)$ where $\sum_j \lambda_j = 1$, $\lambda > 0$ and all $(x_j, y_j) \in P$. This is possible if and only if $\sum_j \lambda_j x_j \in P_1$ and $\sum_j \lambda_j y_j \in P_2$, or, in other words, if and only if $x' \in conv(P_1)$ and $y' \in conv(P_2)$. ∎

Lemma 3. *Let $P \subset \mathbb{R}^n$ be a given closed, pointed set and let $P' = \{(w,x) \in \mathbb{R}^{n+1} \;:\; w \geq ax,\; x \in P\}$ where $a \in \mathbb{R}^n$.*
(i) If P is integral with respect to I, then P' is integral with respect to I.
(ii) $\text{conv}(P') = P''$ *where* $P'' = \{(w,x) \in \mathbb{R}^{n+1} \;:\; w \geq ax,\; x \in \text{conv}(P)\}$.

Proof. (i) Let $p' = (w', x')$ be an extreme point of P'. Clearly, $w' = ax'$, otherwise $p' = 1/2(ax', x') + 1/2(ax' + 2(w' - ax'), x')$ and therefore it can not be extreme.

If x' is an extreme point of P, then x' and therefore p' is integral. On the other hand, if x' is not an extreme point of P, then there exists two distinct points $x^1, x^2 \in P$ such that $x' = 1/2x^1 + 1/2x^2$. In this case $p' = 1/2(ax^1, x^1) + 1/2(ax^2, x^2)$ where $(ax^1, x^1), (ax^2, x^2) \in P'$ and therefore p' can not be extreme.

(ii) Let $p = (\bar{w}, \bar{x}) \in \operatorname{conv}(P')$ and therefore $(\bar{w}, \bar{x}) = \sum_j \lambda_j (w_j, x_j)$ where $\sum_j \lambda_j = 1$, $\lambda > 0$ and $(w_j, x_j) \in P'$ for all j. As $(w_j, x_j) \in P'$, $x_j \in P$ for all j. Therefore $\sum_j \lambda_j (a x_j, x_j) = (a\bar{x}, \bar{x}) \in P''$ and as $\bar{w} \geq a\bar{x}$, we have $(\bar{w}, \bar{x}) \in P''$.

Conversely, assume $p = (\bar{w}, \bar{x}) \in P''$. As $\bar{x} \in \operatorname{conv}(P)$, $\bar{x} = \sum_j \lambda_j x_j$ where $x_j \in P$ and $\sum_j \lambda_j = 1$, $\lambda > 0$. In this case, clearly $\sum_j \lambda_j (a x_j, x_j) = (a\bar{x}, \bar{x}) \in \operatorname{conv}(P')$ and therefore $(\bar{w}, \bar{x}) \in \operatorname{conv}(P')$ as $\bar{w} \geq a\bar{x}$. ∎

We are now ready to present the convex hull of $\bar{Q}$. Let

$$\bar{Q}^c = \Big\{ (w, x, y, z) \in \mathbb{R}^{3n+1} : w \geq \sum_i q_i y_i, \ (x_i, y_i, z_i) \in S_i^c, \ i = 1, 2, \ldots, n \Big\}.$$

Lemma 4. *The set $\bar{Q}^c$ is integral with respect to the indices of z variables. Furthermore,* $\operatorname{conv}(\bar{Q}) = \bar{Q}^c$.

Proof. Let $D = \{(x, y, z) \in \mathbb{R}^{3n} : (x_i, y_i, z_i) \in S_i, \quad i = 1, 2, \ldots, n\}$ so that $\bar{Q} = \{(w, x, y, z) \in \mathbb{R}^{3n+1} : w \geq \sum_{i=1}^n q_i y_i, \ (x, y, z) \in D\}$. By Lemma 3, the convex hull of $\bar{Q}$ can be obtained by replacing D with its convex hull in this description. By Lemma 2, this can simply be done by taking convex hulls of S_i's, that is, by replacing S_i with $\operatorname{conv}(S_i)$ in the description of D. Finally, by Lemma 3, $\bar{Q}^c$ is integral. ∎

2.3. Convex hull description in the original space

In the previous section we presented an extended formulation for the set Q using additional variables. For computational efficiency, it is often desirable to obtain a formulation in the original space so that auxiliary variables are not required. We are therefore interested in projecting the set $\bar{Q}^c$ into the space of (w, x, z). One natural attempt to obtain this projection is to substitute the term x_i^2/z_i for each variable y_i, resulting in the inequality $w \geq \sum_i q_i x_i^2/z_i$. This formula, however, is not suitable for computation as it is not defined for $z_i = 0$. We next present an explicit description of the projection that uses an an exponential number of inequalities. Due to the size of this projection, we conclude that it is more advantageous to work in the extended space for computing purposes. Let

$$Q^c = \Big\{ (w, x, z) \in \mathbb{R}^{2n+1} : w \prod_{i \in S} z_i \geq \sum_{i \in S} q_i x_i^2 \prod_{l \in S \setminus \{i\}} z_l, \ S \subseteq \{1, 2, \ldots, n\} \quad (\Pi)$$
$$u_i z_i \geq x_i \geq l_i z_i, \quad x_i \geq 0, \quad i = 1, 2, \ldots, n \Big\}$$

Notice that a given point $\bar{p} = (\bar{w}, \bar{x}, \bar{z})$ satisfies the nonlinear inequalities in the description of Q^c for a particular $S \subseteq \{1, 2, \ldots, n\}$ if and only if one of the following conditions hold: (*i*) $\bar{z}_i = 0$ for some $i \in S$, or, (*ii*) if all $z_i > 0$, then $\bar{w} \geq \sum_{i \in S} q_i \bar{x}_i^2/\bar{z}_i$. Based on this observation we next show that these (exponentially many) inequalities are sufficient to describe the convex hull of Q in the space of the original variables.

Lemma 5. $Q^c = \text{proj}_{(w,x,z)}(\bar{Q}^c)$.

Proof. Let $\bar{p} = (\bar{w}, \bar{x}, \bar{y}, \bar{z}) \in \bar{Q}^c$ and define $S(\bar{p}) = \{i : z_i > 0\}$. Clearly $u_i\bar{z}_i \geq \bar{x}_i \geq l_i\bar{z}_i$ and $\bar{x}_i \geq 0$ for all $i = 1, 2, \ldots, n$. Furthermore, inequality (Π) is satisfied for all S such that $S \not\subseteq S(\bar{p})$. In addition, notice that, as $q \geq 0$,

$$\bar{w} \;\geq\; \sum_{i \in S(\bar{p})} q_i \bar{y}_i \;\geq\; \sum_{i \in S(\bar{p})} q_i \bar{x}_i^2/\bar{z}_i \;\geq\; \sum_{i \in S'} q_i \bar{x}_i^2/\bar{z}_i$$

for all $S' \subseteq S(\bar{p})$. Therefore $\bar{p}$ satisfies inequality (Π) for all S and $\text{proj}_{(w,x,z)}(\bar{Q}^c) \subseteq Q^c$. Next, let $\bar{p} = (\bar{w}, \bar{x}, \bar{z}) \in Q^c$ be given and let $\bar{y}_i$ be 0 if $z_i = 0$ and $\bar{x}_i^2/\bar{z}_i$, otherwise. It is easy to see that $(\bar{x}_i, \bar{y}_i, \bar{z}_i) \in S_i$ for all $i \in \{1, 2, \ldots, n\}$. Furthermore,

$$\bar{w} \;\geq\; \sum_{i \in S(\bar{p})} q_i \bar{x}_i^2/\bar{z}_i \;=\; \sum_{i \in S(\bar{p})} q_i \bar{y}_i \;=\; \sum_{i=1}^{n} q_i \bar{y}_i$$

implying that $(\bar{w}, \bar{x}, \bar{y}, \bar{z}) \in \bar{Q}^c$ and therefore $Q^c \subseteq \text{proj}_{(w,x,z)}(\bar{Q}^c)$. ∎

Also note that all of the exponentially many inequalities that are used in the description of Q^c are indeed necessary. To see this, consider a simple instance with $u_i = l_i = q_i = 1$ for all $i \in I = \{1, 2, \ldots, n\}$. For a given $\bar{S} \subseteq I$, let $p^{\bar{S}} = (\bar{w}, \bar{x}, \bar{z})$ where $\bar{w} = |\bar{S}| - 1$, $\bar{z}_i = 1$ if $i \in \bar{S}$, and $\bar{z}_i = 0$ otherwise, and $\bar{x} = \bar{z}$. Note that $p^{\bar{S}} \notin Q^c$. As $\bar{z}_i = q_i\bar{x}_i^2$, inequality (Π) is satisfied by $\bar{p}$ for $S \subseteq I$ if and only if

$$(|\bar{S}| - 1) \prod_{i \in S} \bar{z}_i \geq |S| \prod_{i \in S} \bar{z}_i.$$

Note that unless $S \subseteq \bar{S}$, the term $\prod_{i \in S} \bar{z}_i$ becomes zero and therefore inequality (Π) is satisfied. In addition, inequality (Π) is satisfied whenever $|\bar{S}| > |S|$. Combining these two observations, we can conclude that the only inequality violated by $p^{\bar{S}}$ is the one with $S = \bar{S}$.

2.4. SOCP Representation

A second-order cone constraint is a constraint of the form

$$\|Ax + b\|_2 \leq c^T x + d. \tag{5}$$

The set of points x that satisfy (5) forms a convex set, and efficient and robust algorithms exist for solving optimization problems containing second-order cone constraints [33, 27]. An interesting and important observation from a computational standpoint is that the nonlinear inequalities in the definitions of the sets S^c and $\bar{Q}^c$ can be written as second-order cone constraints. All the nonlinear constraints in the definition S^c and $\bar{Q}^c$ are of the simple form

$$x^2 \leq yz \text{ with } y \geq 0, z \geq 0, \tag{6}$$

and this is algebraically equivalent to the second-order cone constraint

$$\|(2x, y-z)^T\| \leq y+z. \tag{7}$$

Constraints of the form (6) are often called *rotated second order cone* constraints. The computational benefit of dealing with inequalities (6) as second-order cone constraints rather than general nonlinear constraints will be demonstrated in Section 5.1.

3. The Convex Hull of the Union of a Point and a Convex Set

We next extend the observations presented in Section 2 to describe the convex hull of a point $\bar{x} \in \mathbb{R}^n$ and a bounded convex set defined by analytic functions. In other words, using an indicator variable $z \in \{0,1\}$, define $W^0 = \big\{(x,z) \in \mathbb{R}^{n+1} \ : \ \ x = \bar{x},\ z = 0\big\}$, and

$$W^1 = \big\{(x,z) \in \mathbb{R}^{n+1} \ : \ f_i(x) \leq 0 \text{ for } i \in I,\ u \geq x - \bar{x} \geq l,\ z = 1\big\}$$

where $u, l \in \mathbb{R}^n_+$, and $I = \{1, \ldots, t\}$. We are interested in the convex hull of $W = W^1 \cup W^0$. Clearly, both W^0 and W^1 are bounded and W^0 is a convex set. Furthermore, if W^1 is also convex then

$$\text{conv}(W) = \{p \in \mathbb{R}^{n+1} \ : \ p = \alpha p^1 + (1-\alpha)p^0,\ p^1 \in W^1,\ p^0 \in W^0,\ 1 \geq \alpha \geq 0\}.$$

We next present a description of conv(W) in the space of original variables. To simplify notation we assume that $\bar{x} = 0$ in the remainder of this section. Note that there is no loss of generality as this is an affine transformation. We next write the description of conv(W) in open form

$$\begin{aligned}\text{conv}(W) \ = \ \Big\{(x,z) \in \mathbb{R}^{n+1} \ : \quad & 1 \ \geq \ \alpha \ \geq \ 0,\ x^0 = 0,\ z^0 = 0,\ z^1 = 1 \\ & x = \alpha x^1 + (1-\alpha)x^0, \quad z = \alpha z^1 + (1-\alpha)z^0, \\ f_i(x^1) \ & \leq 0 \text{ for } i \in I, \quad u \geq x^1 - \bar{x} \geq l \ \Big\}.\end{aligned} \tag{XF}$$

The additional variables used in this description can be projected out to obtain a description in the space of the original variables.

Lemma 6. *If* W^1 *is convex, then* conv$(W) = W^- \cup W^0$*, where*

$$W^- \ = \ \Big\{(x,z) \in \mathbb{R}^{n+1} : \ f_i(x/z) \leq 0 \textit{ for } i \in I, \quad uz \geq x \geq lz, \quad 1 \geq z > 0\Big\}.$$

Proof. As x^0, z^0 and z^1 are fixed in (XF), it is possible to substitute out these variables. In addition, as $z = \alpha$ after these substitutions, we can eliminate α. Furthermore, as $x = \alpha x^1 = z x^1$, we can eliminate x^1 by replacing it with x/z provided that $z > 0$. If, on the other hand, $z = 0$, clearly $(x, 0) \in$ conv(W) if and only if $(x, 0) \in W^0$. ∎

We next show that W^0 is contained in the closure of W^-.

Lemma 7. *For* $1 \geq z > 0$*, let* $Q^c(z) = \Big\{x \in \mathbb{R}^n \ : \ f_i(x/z) \leq 0 \text{ for } i \in I, \quad uz \geq x \geq lz\Big\}$*. If all* $f_i(x)$ *are bounded in* $[l,u]$*, then,*

$$lim_{z \to 0^+} \ Q^c(z) = \Big\{x \in \mathbb{R}^n \ : \quad x = 0\Big\}$$

Proof. Let $\{z_k\} \subset (0,1)$ be a sequence converging to 0. As, by definition, $Q^c(z) \neq \emptyset$ for $z \in (0,1)$, there exists a corresponding sequence $\{x_k\}$ such that $x_k \in Q^c(z_k)$. Clearly, $uz \geq x_k \geq lz$ and therefore $\{x_k\}$ converges to 0. ∎

Combining the previous lemmas, we obtain the following result.

Corollary 1. $\mathrm{conv}(W) \ = \ \mathrm{closure}(W^-)$.

We would like to emphasize that even when $f(x)$ is a convex function $f_i(x/z)$ may not be convex. However, for $z > 0$ we have

$$f_i(x/z) \leq 0 \quad \Leftrightarrow \quad z^t f_i(x/z) \leq 0 \tag{8}$$

for any $t \in \mathbb{R}$. In particular, taking $t = 1$ gives $zf_i(x/z)$ which is known to be convex provided that $f(x)$ is convex. We discuss this further in Section 4.1. We also note that if $f(x)$ is SOCP-representable, then $zf_i(x/z)$ is also SOCP-representable and in particular, if W^1 is defined by SOCP-representable functions, then so is $\mathrm{conv}(W)$. We will show the benefits of employing SOC solvers for (non-quadratic) SOC-representable sets in Section 5.2.

We next show that when all $f_i(x)$ that define W^1 are polynomial functions, convex hull of W can be described explicitly.

Lemma 8. *Let* $f_i(x) = \sum_{t=1}^{p_i} c_{it} \prod_{j=1}^n x_j^{q_{itj}}$ *for all* $i \in I$*. Let* $q_{it} = \sum_{j=1}^n q_{itj}$*,* $q_i = \max_t\{q_{it}\}$ *and* $\bar{q}_{it} = q_i - q_{it}$*. If all* $f_i(x)$ *are convex and bounded in* $[l,u]$*, then* $\mathrm{conv}(W) = W^c$*, where*

$$W^c = \Big\{(x,z) \in \mathbb{R}^{n+1} : \ \sum_{t=1}^{p_i} c_{it} z^{\bar{q}_{it}} \prod_{j=1}^{n} x_j^{q_{itj}} \leq 0 \text{ for } i \in I, \ zu \geq x \geq lz, \ 1 \geq z \geq 0\Big\}.$$

Proof. Note that $f_i(x/z) = \sum_{t=1}^{p_i} c_{it} z^{-q_{it}} \prod_{j=1}^n x_j^{q_{itj}}$. Therefore, multiplying $f_i(x/z) \leq 0$ by z^{q_i}, one obtains the expression above. Clearly, $W^c \cap \{z > 0\} = W^-$ and $W^c \cap \{z = 0\} = W^0$. ∎

4. Connections to Earlier Work

We next relate our results to earlier works that have appeared in the literature.

4.1. Convex Hulls of the Union of Convex Sets

Given a collection of bounded convex sets, it is easy to define an extended formulation to describe their convex hull using additional variables, similar to (XF). Producing a description in the space of original variables, however, appears to be very hard. The particular case we considered in the previous section involves only two sets, one of which consists of a single point. For the sake of completeness we next summarize some related results from Ceria and Soares [12].

Ceria and Soares [12] use *perspective* functions of the functions that define the original sets to produce an extended formulation for the convex hull description. If the original sets are defined by convex functions, their perspective functions are also convex. More precisely, for $t = 1, \ldots, p$, let $G^t : \mathbb{R}^n \to \mathbb{R}^{m_t}$ be a mapping defined by convex functions and assume that the corresponding set

$$K^t = \{x \in \mathbb{R}^n \ : \ G^t(x) \leq 0\}$$

is bounded. Let $\tilde{G}^t : \mathbb{R}^{n+1} \to \mathbb{R}^{m_t}$ be the perspective mapping defined as

$$\tilde{G}^t(\lambda, x) = \begin{cases} \lambda G^t(x/\lambda) & \text{if } \ \lambda > 0 \\ 0 & \text{if } \ \lambda = 0 \\ \infty & \text{otherwise} \end{cases}$$

We next state a important observation from Ceria and Soares [12] that shows the use of perspective functions to obtain convex hulls of convex sets.

Lemma 9 ([12]). *Let K^t be defined as above for $t \in T$, and let $K = \mathrm{conv}(\cup_{t\in T} K^t)$. Then, $x \in K$ if and only if the following nonlinear system is feasible:*

$$x = \sum_{t\in T} x^t; \quad \sum_{t \in T} \lambda_t = 1; \quad \tilde{G}^t(\lambda_t, x^t) \leq 0, \quad \lambda_t \geq 0, \quad \forall t \in T.$$

Furthermore, all $\tilde{G}^t$ are convex mappings provided that all G^t are convex.

Therefore, our observations in Section 3 specialize Lemma 9 to the case when $|T| = 2$ and one of the sets contain a single point. In this special case we show that a description of the convex hull in the original space can be obtained easily.

4.2. Perspective Cuts

Building on the work of Ceria and Soares [12], Frangioni and Gentile [14] introduce the class of *perspective cuts* for mixed integer programs of the form

$$\min_{(x,z)\in\mathbb{R}^n\times\mathbb{B}} \Big\{ f(x) + cz \mid Ax \leq bz \Big\},$$

where (i) $X = \{x \mid Ax \leq b\}$ is bounded (also implying $\{x \mid Ax \leq 0\} = \{0\}$), (ii) $f(x)$ is a convex function that is finite on X, and (iii) $f(0) = 0$. Under these assumptions, they show that for any $\bar{x} \in X$ and $s \in \partial f(\bar{x})$, the *perspective cut*

$$v \geq f(\bar{x}) + c + s^T(x - \bar{x}) + (c + f(\bar{x}) - s^T\bar{x}))(z - 1) \tag{9}$$

is valid for the equivalent mixed integer program

$$\min_{(x,z,v)\in\mathbb{R}^n\times\mathbb{B}\times\mathbb{R}} \Big\{ v \mid v \geq f(x) + cz, Ax \leq bz \Big\}.$$

Frangioni and Gentile [14] derive the linear inequalities (9) from a first-order analysis of the convex envelope of the perspective function of $f(x)$. A similar first-order argument can be used to derive inequality (9) from the characterization of the convex hull of the union of a convex set and a point given in Section 3. First define $P^0 \stackrel{\text{def}}{=} \big\{(x, z, v) \in \mathbb{R}^{n+2} :\ x = 0, z = 0, v = 0\big\}$, and

$$P^1 \stackrel{\text{def}}{=} \big\{(x, z, v) \in \mathbb{R}^{n+2} : Ax \leq b,\ f(x) + c - v \leq 0,\ u_x \geq x \geq l_x,\ u_v \geq v \geq l_v,\ z = 1\big\}$$

where bounds on variables x and v are introduced without loss of generality. Corollary 1 states that $\text{conv}(P^0 \cup P^1)$ is the closure of

$$P^- \stackrel{\text{def}}{=} \Big\{(x, z, v) \in \mathbb{R}^{n+2} \mid Ax \leq b,\ zf(x/z) + cz - v \leq 0, u_x z \geq x \geq l_x z, \\ u_v z \geq v \geq l_v z,\ 1 \geq z \geq 0\Big\}.$$

For any $\bar{z} > 0$, a first-order (outer)-approximation of the nonlinear constraint $zf(x/z) + cz - v \leq 0$ about the point $(\bar{x}, \bar{z}, \bar{v})$ gives

$$0 \geq \bar{z}f(\bar{x}/\bar{z}) + c\bar{z} - \bar{v} + \begin{bmatrix} s \\ (-1/\bar{z})\bar{x}^T s_{x/z} + f(\bar{x}/\bar{z}) + c \\ -1 \end{bmatrix}^T \begin{bmatrix} x - \bar{x} \\ z - \bar{z} \\ v - \bar{v} \end{bmatrix},$$

where $s \in \partial f(\bar{x})$ and $s_{x/z} \in \partial f(\bar{x}/\bar{z})$. Taking $\bar{z} = 1$, $\bar{v} = f(\bar{x})+c$, and rearranging terms gives inequality (9) above.

The implication of this analysis is that the perspective cuts of Frangioni and Gentile [14] are outer approximation cuts for $\text{conv}(P^0 \cup P^1)$. Thus, the strength of the perspective relaxation is equivalent to that of adding all (infinitely-many) perspective cuts to the formulation. The disadvantage of the perspective reformulation over perspective cuts is that the inequalities used in the reformulation are nonlinear. We discuss a direct computational comparison between the nonlinear perspective reformulation and perspective cuts in Section 5.3.

5. Applications

In this section, three applications are described: a quadratic-cost uncapacitated facility location problem recently studied by Günlük et al. [20], a network design problem under queuing delay, first discussed by Boorstyn and Frank [10], and a portfolio optimization problem with minimum buy-in thresholds [29, 7, 22]. In each case, the positive impact of the perspective reformulation and the ability to model the nonlinear inequalities in the reformulations as second-order cone constraints is demonstrated.

5.1. Separable Quadratic UFL

The Separable Quadratic Uncapacitated Facility Location Problem (SQUFL) was introduced by Günlük et al. [20]. In the SQUFL, there is a set of customers N, a set of facilities M and there is a fixed cost c_i for opening a facility $i \in M$. All customers have unit demand that can be satisfied using open facilities only. The shipping cost is proportional to the square of the quantity delivered. Letting z_i indicate if facility $i \in N$ is open, and x_{ij} denote the fraction of customer j's demand met from facility i, SQUFL can be formulated as follows:

$$\begin{aligned}
\min \quad & \sum_{i\in M} c_i z_i + \sum_{i\in M}\sum_{j\in N} q_{ij} x_{ij}^2 \\
\text{subject to} \quad & x_{ij} \le z_i \quad \forall i \in M, \forall j \in N, \\
& \sum_{i\in M} x_{ij} = 1 \quad \forall j \in N, \\
& z_i \in \{0,1\}, \qquad x_{ij} \ge 0 \quad \forall i \in M, \forall j \in N.
\end{aligned}$$

To apply the perspective formulation, auxiliary variables y_{ij} are used to replace the terms x_{ij}^2 in the objective function. In addition the following constraints

$$x_{ij}^2 - y_{ij} \le 0 \quad \forall i \in M, j \in N, \tag{10}$$

$$y_{ij} \le z_i \quad \forall i \in M, j \in N, \tag{11}$$

are added. In this reformulation, if $z_i = 0$, then $x_{ij} = y_{ij} = 0 \ \forall j \in N$, while if $z_i = 1$, the constraints (10) define the set of feasible points. Therefore, we can strengthen the formulation using the perspective counterparts of constraints (10)

$$x_{ij}^2 - z_i y_{ij} \le 0 \qquad \forall i \in M, \forall j \in N. \tag{12}$$

5.1.1. Generating the Test Set We generated random instances similar to the ones in [20]. For each facility $i \in M$, a location p_i is generated uniformly in $[0,1]^2$ and the variable cost parameter was calculated as $q_{ij} = 50\|p_i - p_j\|_2$. The fixed cost c_i of opening a facility is generated uniformly in $[1, 100]$. Ten instances were created for each $m \in \{10, 20, 30, 40\}$ and $n \in \{30, 50, 100, 200\}$.

5.1.2. Computational Results with an Interior Point Solver. In Table 1 we summarize our results with the open-source MINLP solver `BONMIN` [9] using `Ipopt` [35] as the NLP solver on a 1.8GHz AMD Opteron CPU. In Table 1 $\bar{z}_R$ and $\bar{z}_P$ represent the average value of the continuous relaxation of the original formulation, and the perspective reformulation respectively and $\bar{z}^*$ is the average value of the optimal solution. The table also displays the number of instances out of 10 (# Sol.) that were solved within a time limit of 8 hours, the average number of nodes ($\bar{N}$) required to solve the instances, and the average CPU time ($\bar{T}$) in seconds for both the original and perspective formulations.

First note that nearly all of the integrality gap is closed at the root node by the perspective reformulation and consequently, the number of nodes needed

Table 1. Relaxation Values and Solution Times for SQUFL

					Original Formulation			Perspective Formulation		
m	n	$\bar{z}_R$	$\bar{z}_P$	$\bar{z}^*$	# Sol.	$\bar{N}$	$\bar{T}$	# Sol.	$\bar{N}$	$\bar{T}$
10	30	105.8	196.5	197.9	10	333	8.9	10	15	3.7
10	50	160.4	312.6	314.6	10	406	18.0	10	11	4.9
10	100	266.5	460.4	462.0	10	441	36.7	10	9	7.7
10	200	470.7	733.6	737.0	10	350	59.7	10	7	15.2
20	30	81.7	185.3	185.6	10	3452	213.7	10	37	39.9
20	50	111.6	274.8	276.2	10	5526	601.4	10	31	85.9
20	100	166.3	412.7	414.5	7	25901	12263.9	10	35	677.1
20	200	283.5	650.8	653.1	0	-	-	10	27	1925
30	30	64.1	157.8	159.4	9	17837	1822.7	10	62	192.8
30	50	82.1	241.6	243.3	1	61062	23760.2	10	56	650.3
30	100	126.0	343.4	345.6	0	-	-	10	51	4565.4
30	200	200.7	545.8	547.4	0	-	-	9	44	16858.5
40	30	58.6	146.4	147.7	7	55660	9319.6	10	71	224.3
40	50	74.1	198.7	200.0	0	-	-	10	85	3030.6
40	100	109.6	309.8	311.2	0	-	-	10	64	8420.8
40	200	161.4	478.3	-	0	-	-	0	-	-

to solve the problem is orders of magnitude smaller. Also notice that CPU time per node increases dramatically when the perspective formulation is applied. For example, for $n = 30$ and $m = 200$, `Ipopt` takes, on average, 383 CPU seconds to evaluate a node. As a general the interior-point solver, `Ipopt` is does not exploit the special second-order cone structure of the perspective reformulation. Furthermore, as the functions $(x^2 - yz)$ that appear in the reformulation are not convex, `Ipopt` may only guarantee convergence to a stationary point (not the globally optimal solution to the NLP relaxation). Thus, it is possible that he final solution produced by `Ipopt/BONMIN` would not be a true optimal solution to the instance. Further, the solution `Ipopt` converges to is highly dependent on the initial iterate provided. For the experiments reported in Table 1 a starting point of $x_{ij} = 1/m \, \forall i, j$, $z_i = 1/m \, \forall i$, and $y_{ij} = 1/(m^2) \forall i, j$ was used at the root node. Interestingly, using this starting point, `BONMIN` never incorrectly pruned a node as a result of a suboptimal solution reported by `Ipopt`.

5.1.3. Computational Results with a SOCP Solver. We also tried the SOCP solver Mosek (version 5.0) [27] to solve the perspective reformulation where the nonlinear inequalities are represented in second-order-cone form. Table 2 shows the number of nodes (N) and CPU seconds (T) required by Mosek to solve random instances of various sizes. The table also shows the time per node (T/N) when possible. Note that the speed-up is solely due to the reduced time to solve relaxations at nodes. In addition, larger instances (up to size $n = 50$, $n = 200$) can be solved.

As pointed out by a referee, `Ipopt` may be more effective at solving nonlinear programs where all constraint functions are convex. Therefore, the test instances were run again replacing the inequalities $x_{ij}^2 - y_{ij} z_i \leq 0 \; \forall i \in M, \forall j \in N$, with

$$\|(2x_{ij}, y_{ij} - z_i)^T\| \leq y_{ij} + z_i \qquad \forall i \in M, \forall j \in N.$$

Table 2. Solution Times for SOC-Perspective Reformulation of SQUFL

m	n	T	N	T/N(SOCP)	T/N(NLP)
20	100	3.8	12	0.3	19.3
20	200	9.6	11	0.9	71.3
30	100	9.6	30	0.3	89.5
30	200	141.9	63	2.3	383.1
40	100	76.4	54	1.4	131.6
40	200	101.3	45	2.3	-
50	100	61.6	49	1.3	-
50	200	140.4	47	3.0	-

In general, this reformulation technique did improve the performance of `Ipopt`. The solution of the NLP relaxations was on average nearly 7 times faster with the convex reformulation. However, the reader will note that these times are still significantly slower than those obtained by a specialized solver for SOCP.

Günlük et al. [20] also derive cutting planes to strengthen the continuous relaxation of the SQUFL. The first part of Table 3 is taken from their paper where z_R and z_{GLW} denote the value of the original and strengthened root relaxations respectively. The second part of the table gives the value of the root relaxation of the perspective reformulation (z_P), and the optimal solution value (z^*). Clearly, the perspective reformulation performs significantly better than their cutting planes. The largest instance in Table 3 was solved by Lee [24] using `BONMIN`. The solution required 16,697 CPU seconds and 45,901 nodes for the original formulation, and 21,206 CPU seconds and 29,277 nodes for the strengthened formulation. The same instance was solved in *23 CPU seconds* on the same machine type (enumerating 44 nodes) using Mosek on the perspective reformulation. The speedup factor is more than 700.

Table 3. Comparison of Relaxation Bounds for SQUFL

m	n	z_R	z_{GLW}	z_P	z^*
10	30	140.6	326.4	346.5	348.7
15	50	141.3	312.2	380.0	384.1
20	65	122.5	248.7	288.9	289.3
25	80	121.3	260.1	314.8	315.8
30	100	128.0	327.0	391.7	393.2

5.1.4. Commercial MINLP solvers We also tried the DICOPT [23] and BARON [34] solvers on the original formulation. Within a time limit of eight hours per instance, DICOPT was only able to successfully solve small instances (up to size $n = 10$, $m = 100$), whereas BARON was not able to solve any of the instances in our test suite. Clearly, automatically recognizing and exploiting the perspective reformulation could significantly improve the computational performance of MINLP software.

5.2. Network Design with Congestion Constraints

The next application is a network design problem with requirements on queuing delay. Similar models appear in [10], [6], and [11]. In the problem, there is a set of commodities K to be shipped over a capacitated directed network $G = (N, A)$. The capacity of arc $(i, j) \in A$ is u_{ij}, and each node $i \in N$ supplies or demands a specified amount b_i^k of commodity k. There is a fixed cost c_{ij} of opening each arc $(i, j) \in A$, and we introduce {0,1}-variables z_{ij} to indicate whether arc $(i, j) \in A$ is opened. The quantity of commodity k routed on arc (i, j) is measured by variable x_{ij}^k and $f_{ij} = \sum_{k \in K} x_{ij}^k$ denotes the total flow on the arc. A typical measure of the total weighted congestion (or queuing delay) is

$$\rho(f) \stackrel{\text{def}}{=} \sum_{(i,j)\in A} r_{ij} \frac{f_{ij}}{1 - f_{ij}/u_{ij}},$$

where $r_{ij} \geq 0$ is a user-defined weighting parameter for each arc. We use a decision variables y_{ij} to measure the contribution of the congestion on arc (i, j) to the total congestion $\rho(f)$. The network should be designed so as to keep the total queuing delay less than a given value β, and this is to be accomplished at minimum cost. The resulting optimization model (NDCC) can be written as

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij} z_{ij} & & \\
\text{subject to} \quad & \sum_{(j,i)\in A} x_{ij}^k - \sum_{(i,j)\in A} x_{ij}^k = b_i^k & \forall i \in N,\ \forall k \in K, & \\
& \sum_{k\in K} x_{ij}^k - f_{ij} = 0 & \forall (i,j) \in A, & \\
& f_{ij} \leq u_{ij} z_{ij} & \forall (i,j) \in A, & \quad (13) \\
& y_{ij} \geq \frac{r_{ij} f_{ij}}{1 - f_{ij}/u_{ij}} & \forall (i,j) \in A, & \quad (14) \\
& \sum_{(i,j)\in A} y_{ij} \leq \beta, & & \\
& z_{ij} \in \{0,1\} & \forall (i,j) \in A, & \\
& x \in \mathbb{R}_+^{|A|\times|K|},\ y \in \mathbb{R}_+^{|A|},\ f \in \mathbb{R}_+^{|A|}. & &
\end{aligned}$$

An observation not previously made in the literature regarding this problem is that the congestion inequalities (14) can be written as SOC constraints. Multiplying both sides of (14) by $1 - f_{ij}/u_{ij} > 0$, adding $r_{ij} f_{ij}^2$ to both sides of the inequality, and factoring the left-hand-side gives an equivalent constraint

$$(y_{ij} - r_{ij} f_{ij})(u_{ij} - f_{ij}) \geq r_{ij} f_{ij}^2. \qquad (15)$$

Because the inequalities $y_{ij} \geq r_{ij} f_{ij}$ and $u_{ij} \geq f_{ij}$ most hold in any feasible solution, (15) is precisely a constraint in rotated second-order cone form (6).

Cut-set inequalities (see,[25, 8]) are known to strengthen the continuous relaxation of network design problems. In our computational experiments, we used the following most basic and effective cut-set inequalities: Let δ_i denote the total flow originating from node i and $\tau_i \in Z_+$ be such that $\sum_{ij \in A'} u_{ij} < \delta_i$ for all $A' \subset A$ such that $|A'| \leq \tau_i - 1$. The associated cut-set inequity for node i is

$$\sum_{(i,j) \in A} z_{ij} \geq \tau_i.$$

We added these inequalities for both incoming and outgoing arcs for all $i \in N$. More elaborate inequalities could also be added (see [8]), but our goal is to examine the impact of the perspective reformulation, not strong linear inequalities.

In the formulation of NDCC, note that if $z_{ij} = 0$, then the constraints (13) force $f_{ij} = 0$, and the constraints (14) are redundant for the arc (i, j). However, if $z_{ij} = 1$, then the definitional constraint (14) for the corresponding y_{ij} must hold. Therefore, each constraint (14) can be replaced by its perspective counterpart:

$$z_{ij} \left[\frac{r_{ij} f_{ij}/z_{ij}}{1 - f_{ij}/(u_{ij} z_{ij})} - \frac{y_{ij}}{z_{ij}} \right] \leq 0. \tag{16}$$

The constraints (16) can also be written as second order cone constraints in a similar fashion to the non-perspective version (14). Specifically, simplifying the left-hand size of the inequality (16), adding $r_{ij} f_{ij}^2$ to both sides of the simplified inequality and factoring gives the equivalent constraints

$$(y_{ij} - r_{ij} f_{ij})(u_{ij} z_{ij} - f_{ij}) \geq r_{ij} f_{ij}^2,$$

which is a rotated second-order cone constraint since $y_{ij} \geq r_{ij} f_{ij}$ and $u_{ij} z_{ij} \geq f_{ij}$ for any feasible solution. The fact that the inequalities in the perspective reformulation of (14) are SOC-representable is no surprise. In fact, Ben-Tal and Nemirovski [5] (Page 96, Proposition 3.3.2) show that the perspective transformation of a function whose epigraph is a SOC-representable set is (under mild conditions) always SOC-representable.

5.2.1. Computational Results For test instances we created random graphs where each arc is present with probability 0.2. In these instances, each node is the unique source of exactly one commodity. Let $s(k)$ denote the source node for commodity $k \in K$. The demands were created as follows

$$\begin{aligned} b_i^k &= \lceil \mathcal{U}(5, 25) \rfloor & \forall k \in K \ \forall i \in (I \setminus \{s(k)\}), \\ b_{s(k)}^k &= - \sum_{i \in I \setminus \{s(k)\}} b_i^k & \forall k \in K \end{aligned}$$

where $\mathcal{U}(a, b)$ is a uniformly distributed random number in the interval (a, b), and $\lceil x \rfloor$ is the closest integer to x. Let $B = \sum_{k \in K} \sum_{i \in I \setminus \{s(k)\}} b_i^k$ be the total demand. We set $\beta = \kappa B$ where κ is the smallest integer necessary to make the linear relaxation feasible. Finally, for all $(i,j) \in A$ we set $u_{ij} = \lceil \mathcal{U}(1.0, 5.0)B/|A| \rfloor$ $r_{ij} = 1.0$ and $c_{ij} = \mathcal{U}(1,4)$.

All of the instances were created in the GAMS modeling language and solved using the branch-and-bound mixed integer SOCP code of Mosek. A time limit of 4 hours was imposed on each instance. We created graphs for $|N| = 20$ and $|N| = 30$. Results comparing the two formulations is presented in Table 4. Notice that the perspective reformulation helps the solvability considerably. Of the 35 instances of size $|N| = 20$, 2 can be solved within the time limit with the original formulation, and 29 can be solved with the perspective reformulation. Of the 6 that don't solve, 4 fail due to numerical difficulties with solving the relaxation, and 2 hit the time limit. Of the 35 instance of size $|N| = 30$, neither the original formulation nor perspective formulation are able to solve any of these instances. However, the average remaining optimality gap after 4 hours was 57.1% for the original formulation and 7.03% for the perspective formulation.

Table 4. Impact of Perspective Reformulation on Network Design Instances.

	Original Formulation					Perspective Formulation				
N	# Sol.	z_{root}	z_L	z_U	*Nodes*	# Sol.	z_{root}	z_L	z_U	*Nodes*
20	2	80.6	144.1	183.9	30086.2	26	165.3	178.4	179.3	11347.4
30	-	150.7	261.3	392.2	3861.1	-	353.6	355.7	379.2	5375.8

5.3. Mean-Variance Optimization

A canonical optimization problem in financial engineering is to find a minimum variance portfolio that meets a minimum return requirement [26]. In the problem, there is a set N of assets available for purchase. The expected return of asset $i \in N$ in given by α_i, and the covariance of the returns between every pair of assets is given in the form a positive-definite matrix $Q \in \mathbb{R}^{n \times n}$. The canonical problem is often augmented with a number of business rules that require the introduction of binary variables in straightforward optimization models. For example, there may be minimum (ℓ_i) and maximum (u_i) buy-in thresholds for each asset $i \in N$, resulting the the following optimization problem (MVOBI):

$$\min\{x^T Q x \mid e^T x = 1, \alpha^T x \geq \rho, \ell_i z_i \leq x_i \leq u_i z_i \ \forall i \in N\}, \tag{17}$$

where the decision variable x_i is the percentage of the portfolio invested in asset i and z_i is a binary variable indicating the purchase of asset i. Imposing a cardinality constraint on the number of different assets purchased can be achieved by adding a constraint $\sum_{i \in N} z_i \leq K$. Unfortunately, direct application of the perspective reformulation to (17) is not possible, as the objective is not a separable function of the decision variables x.

However, in many practical applications, the covariance matrix is obtained from a *factor model* and has the form $Q = B\Omega B^T + \Delta^2$, for a given *exposure* *matrix*, $B \in \mathbb{R}^{n \times f}$, positive-definite *factor-covariance matrix* $\Omega \in \mathbb{R}^{f \times f}$, and positive definite, diagonal *specific-variance* matrix $\Delta \in \mathbb{R}^{n \times n}$ [29]. If a factor model is given, a separable portion of the objective function is easily extracted by introducing variables y_i, changing the objective to

$$\min x^T (B\Omega B^T)x + \sum_{i \in N} \Delta_{ii} y_i,$$

and enforcing the constraints $y_i \geq x_i^2 \ \forall i \in N$.

Even if the covariance matrix Q does not directly have embedded diagonal structure from a factor model, then, as suggested by Frangioni and Gentile [14], it is still possible to extract a separable component from Q. Specifically, the matrix Q may be decomposed into $Q = R + D$, for some positive, diagonal matrix D such that $R = Q - D$ remains positive-definite. The objective can be changed to $\min x^T Rx + x^T Dx$, and $x^T Dx$ is separable in x. Frangioni and Gentile [14] suggest using $D = \lambda_n I$, where $\lambda_n > 0$ is the smallest eigenvalue of Q. In our computational experiments, we follow their advice and use $D = (\lambda_n - \varepsilon)I$, where $\varepsilon = 0.001$ so that R is strictly positive definite.In subsequent work, Frangioni and Gentile [15] show how "more" of the separable structure of Q can be extracted into D through the solution of a semidefinite program.

In order to solve the instance entirely in a second-order cone programming framework, we use the well-known transformation [5] of a convex quadratic program into a second order cone program. To transform the instance, a Cholesky factorization of $R = MM^T$, is taken, the auxiliary variables $w = M^T x$ are introduced, so that $\|w\| = x^T Rx$.

$$\min \quad v + \sum_{i \in N} D_{ii} y_i \tag{18}$$

$$\text{subject to} \quad w - M^T x = 0$$

$$v - \|w\| \geq 0 \tag{19}$$

$$y_i - x_i^2 \geq 0 \qquad \forall i \in N \tag{20}$$

$$\sum_{i \in N} x_i = 1 \tag{21}$$

$$\sum_{i \in N} \alpha_u x_i \geq \rho \tag{22}$$

$$\ell_i z_i \leq x_i \leq u_i z_i \qquad \forall i \in N \tag{23}$$

The inequalities (19) can easily be placed in rotated second order cone form (6). Since $z_i = 0$ implies that constraint (20) is redundant, and, while $z_i = 1$ implies that we would like the inequality to hold, the perspective reformulation may be applied, replacing the constraints (20) with inequalities

$$y_i z_i - x_i^2 \geq 0 \forall i \in N. \tag{24}$$

The inequalities (24) are precisely in the rotated second order cone form (6), so they can be effectively handled by software such as Mosek.

In Table 5 we summarize computational results on twenty instances of the MVOBI problem (ten instances each for $|N| = 200$ and $|N| = 300$) . These instances were created by Frangioni and Gentile [14], and optimal solutions for the instances are reported at `http://www.di.unipi.it/optimize/Data/MV.html`. Mosek branch-and-bound solver was run on each instance with a time limit of 10,000 CPU seconds. If z_R is the value of the SOCP-relaxation at the root node, z^* is the optimal solution, z_L and z_U are the best lower and upper bounds found by Mosek, the table reports the average root gap to optimal (RGO $= 100(z^* - z_R)/z_R$), the average final gap to optimal (FGO $= 100(z^* - z_L)/z_L$), the average final gap (FG $= 100(z_U - z_L)/z_L$)), and the average number of nodes.

Table 5. Integrality Gaps of Formulations for MVOBI

	Original Formulation				Perspective Formulation			
N	RGO	FGO	FG	Nodes	RGO	FGO	FG	Nodes
200	667.8	181.8	185.1	42879.4	7.0	3.0	4.2	8118.4
300	1179.3	488.8	490.0	45629.9	6.0	3.9	5.9	2460.7

In these experiments, the Mosek conic IP solver was able to solve only one instance to optimality. Nevertheless, the perspective reformulation significantly improves the lower bound. For these instances, the linearization approach of Frangioni and Gentile [14], especially when used in conjunction with their technique for choosing the diagonal matrix D [15] appears to be more effective than the direct perspective formulation.

However, a distinct advantage of the perspective reformulation is that specialized cutting-plane based procedures are not necessary to achieve the improved performance. The reformulation can be simply implemented with a modeling language. We view this result as pointing clearly to the need for improvements in conic IP software.

6. Conclusions

In this work we derive an explicit characterization of the convex hull of the union of a point and a bounded convex set defined by analytic functions. This characterization can be used to produce strong "perspective" reformulations of many practical mixed integer nonlinear programs. We also show that in many cases, the nonlinear inequalities in the perspective reformulation can be cast as second-order cone constraints, a transformation that greatly improves an instance's solvability.

Continuing work has two primary thrusts: (1) Automatic detection of structures to which the perspective transformation can be applied; and (2) Studying additional simple structures occurring in practical MINLPs in the hope of deriving strong relaxations.

Acknowledgements. Author Linderoth would like to acknowledge support from the US Department of Energy under grant DE-FG02-05ER25694, and by IBM, through the faculty partnership program. The authors thank Shabbir Ahmed for bringing Proposition 3.3.2 of [5] to their attention. An extended-abstract of this work has appeared in IPCO 2008 [18].

References

1. K. Abhishek, S. Leyffer, and J. T. Linderoth. FilMINT: An outer-approximation-based solver for nonlinear mixed integer programs. Preprint ANL/MCS-P1374-0906, Mathematics and Computer Science Division, Argonne National Lab, 2006.
2. S. Aktürk, A. Atamtürk, and S. Gürel. A strong conic quadratic reformulation for machine-job assignment with controllable processing times. Technical Report BCOL Research Report 07.01, Industrial Engineering & Operations Research, University of California, Berkeley, April 2007.
3. A. Atamtürk and V. Narayanan. Conic mixed integer rounding cuts. *Mathematical Programming*, 2008. Forthcoming.
4. E. Balas, S. Ceria, and G. Corneujols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:295–324, 1993.
5. A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization.* SIAM, 2001. MPS/SIAM Series on Optimization.
6. D. Bertsekas and R. Gallager. *Data Networks.* Prentice-Hall, Endlewood Cliffs, NJ, 1987.
7. D. Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74:121–140, 1996.
8. D. Bienstock and O. Günlük. Capacitated network design - Polyhedral structure and computation. *ORSA Journal on Computing*, 8:243–260, 1996.
9. P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5:186–204, 2008.
10. R. Boorstyn and H. Frank. Large-scale network topological optimization. *IEEE Transactions on Communications*, 25:29–47, 1977.
11. B. Borchers and J. E. Mitchell. An improved branch and bound algorithm for mixed integer nonlinear programs. *Computers & Operations Research*, 21:359–368, 1994.
12. S. Ceria and J. Soares. Convex programming for disjunctive optimization. *Mathematical Programming*, 86:595–614, 1999.
13. M. T. Cezik and G. Iyengar. Cuts for mixed 0-1 conic programming. *Mathematical Programming*, 104:179–202, 2005.
14. A. Frangioni and C. Gentile. Perspective cuts for a class of convex 0-1 mixed integer programs. *Mathematical Programming*, 106:225–236, 2006.
15. A. Frangioni and C. Gentile. Sdp diagonalizations and perspective cuts for a class of nonseparable miqp. *Operations Research Letters*, 35:181–185, 2007.
16. R. E. Gomory. An algorithm for the mixed integer problem. Technical Report RM-2597, The RAND Corporation, 1960.
17. I. Grossmann and S. Lee. Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational Optimization and Applications*, pages 83–100, 2003.
18. O. Günlük and J. Linderoth. Perspective relaxation of mixed integer nonlinear programs with indicator variables. In A. Lodi, A. Panconesi, and G. Rinaldi, editors, *IPCO 2008: The Thirteenth Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science*, volume 5035, pages 1–16, 2008.
19. O. Günlük and J. Linderoth. Perspective relaxation of mixed integer nonlinear programs with indicator variables. Technical Report RC24694 (W0811-076), IBM Research Division, November 2008.
20. O. Günlük, J. Lee, and R. Weismantel. MINLP strengthening for separaable convex quadratic transportation-cost ufl. Technical Report RC24213 (W0703-042), IBM Research Division, March 2007.
21. H. Horst, P. M. Pardalos, and V. Thoai. *Introduction to Global Optimization.* Kluwer, Dordrecht, 1995.
22. N. J. Jobst, M. D. Horniman, C. A. Lucas, and G. Mitra. Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. *Quantitative Finance*, 1:489–501, 2001.

23. G. R. Kocis and I. E. Grossmann. Computational experience with DICOPT solving MINLP problems in process systems engineering. *Computers and Chemical Engineering*, 13:307–315, 1989.
24. J. Lee. Mixed-integer nonlinear programming: Some modeling and solution issues. *IBM Journal of Research & Development*, 51:489–497, 2007.
25. T. Magnanti and P. Mirchandani. Shortest paths, single origin-destination network design and associated polyhedra. *Networks*, 23:103–121, 1993.
26. H. M. Markowitz. Portfolio selection. *Journal of Finance*, 7:77–91, 1952.
27. mosek. The mosek optimization tools manual. version 5.0 (revision 84), 2008. `www.mosek.com`.
28. G. Nemhauser and L. Wolsey. A recursive procedure for generating all cuts for 0-1 mixed integer programs. *Mathematical Programming*, 46:379–390, 1990.
29. A. F. Perold. Large-scale portfolio optimization. *Management Science*, 30:1143–1160, 1984.
30. I. Quesada and I. E. Grossmann. An LP/NLP based branch–and–bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering*, 16:937–947, 1992.
31. R. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86:515–532, 1999.
32. R. A. Stubbs. *Branch-and-Cut Methods for Mixed 0-1 Convex Programming*. PhD thesis, Northwestern University, December 1996.
33. J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.
34. M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.
35. 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.