RC24694 (W0811-076) November 11, 2008
Mathematics

IBM Research Report

Perspective Reformulations of Mixed Integer Nonlinear Programs with Indicator Variables

Oktay Günlük
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Jeff Linderoth
Department of Industrial and Systems Engineering
University of Wisconsin-Madison
1513 University Avenue
3226 Mechanical Engineering Building
Madison, WI 53706

Oktay Günlük[1] · Jeff Linderoth[2]

Perspective Reformulations of Mixed Integer Nonlinear Programs with Indicator Variables

June 20, 2008

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 (1996) 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 [27, 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.

More precisely, we study MINLPs that are driven by a collection of indicator variables where each indicator variable controls a subset of the decision variables.

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

Department of Industrial and Systems Engineering,University of Wisconsin-Madison, 1513 University Avenue, 3226 Mechanical Engineering Building, Madison, WI 53706, USA, e-mail: linderoth@wisc.edu

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 $i \ne j$ we can have $V_i \cap V_j \ne \emptyset$. 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:

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

where

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

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 $\varGamma_i$ is a convex set. An important observation is that $\varGamma_i$ 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,

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

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.

Indicator-induced MINLPs can be used to model many interesting problems. We study three applications in this paper: the 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 [26, 7, 21]. In addition, certain classes of unit commitment problems for electrical power generation can be formulated as indicator-induced MINLPs [12], and Aktürk et al. [2] give an indicator-induced MINLP for a job-scheduling problem with controllable processing times.

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 [28] explain how the disjunctive cutting planes of Balas et al. [4] can be applied for MINLP, Cezik and Iyengar [14] extend the Gomory cutting plane procedure [17], and Atamtürk and Narayanan [3] extend the mixed integer rounding procedure of Nemhauser and Wolsey [25] 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] for different families of inequalities for a quadratic cost facility location problem. In some cases these inequalities can be used to strengthen the perspective relaxation even further.

Related to this work, Frangioni and Gentile [15] have introduced a class of linear inequalities called *perspective cuts* for 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 [18], who extend the convex hull characterization of Ceria and Soares [13] 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 $\Gamma_i = \{x \in \mathbb{R}^2 \mid x_1^t - x_2 \leq 0,\, u \geq x_1, x_2 \geq 0\}$ for $t \geq 1$. They use this characterization in an algorithm for solving some classes of nonlinear machine scheduling problems.

The remainder of the paper is divided in four 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 [13] and Frangioni and Gentile [15]. 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 following set:

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

where $q_i \in \mathbb{R}_+$ and $u_i, l_i \in \mathbb{R}$ for all $i = 1, 2, \ldots, n$. The set Q appears in a number of non-linear mixed-integer programs as a substructure, and some examples are given in Section 5. To our knowledge, the first convex hull description of Q was stated without proof in the unpublished Ph.D. thesis of Stubbs [29]. 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 $t \geq 0$. The set is shown in Figure 1.

Note that even though $x^2 - yz$ is not a convex function (its Hessian is not positive semi-definite), the set S^c still defines a convex set in $\mathbb{R}^3_+$. To see this first note that S^c is convex if $T^c = \{(x, y, z) \in \mathbb{R}^3 \,:\, yz \geq x^2,\ x, y, z \geq 0\}$ is convex. To see that T^c is convex, let $p_i = (x_i, y_i, z_i) \in T^c$, for $i = 1, 2$, and consider $p_3 = p_1/2 + p_2/2$. We start with showing that

$$y_1 z_2 + y_2 z_1 \geq 2 x_1 x_2 \tag{5}$$

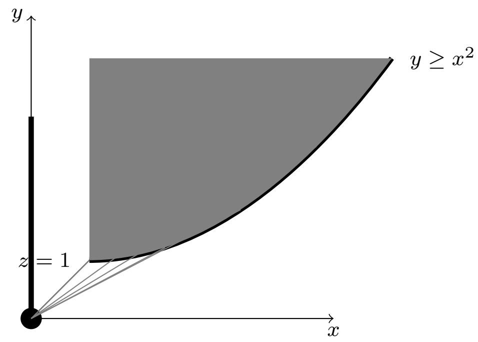

Fig. 1. The set S^c

holds. As $p_1, p_2 \geq 0$, inequality (5) holds when at least one of x_1 or x_2 is zero. Next consider the case when $x_1, x_2 > 0$. To simplify notation, let $a = y_1/x_1$, $b = z_1/x_1$, $c = y_2/x_2$, $d = z_2/x_2$, and note that $a, b, c, d > 0$. As $p_1, p_2 \in T^c$, we have $ab \geq 1$ and $cd \geq 1$ and therefore $b \geq 1/a$ and $c \geq 1/d$. We now write

$$ad + bc - 2 \geq ad + \frac{1}{ad} - 2 = \frac{1}{ad}(ad^2 - 2ad + 1) = \frac{1}{ad}(ad - 1)^2 \geq 0.$$

Therefore, inequality (5) holds when $x_1, x_2 > 0$ as well. Using this property we can now show

$$\begin{aligned} y_3 z_3 = &\ (1/2y_1 + 1/2y_2)(1/2z_1 + 1/2z_2) = 1/4(y_1 z_1 + y_1 z_2 + y_2 z_1 + y_2 z_2) \\ \geq &\ 1/4x_1^2 + 1/2x_1 x_2 + 1/4x_2^2 = (1/2x_1 + 1/2x_2)^2 = x_3^2 \end{aligned}$$

implying that $p^3 \in T^c$, and therefore T^c is convex.

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

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

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

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

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 $p' \in \mathrm{conv}(S)$. In addition, $(0, 1, 0)$ is an (extreme) direction of S^0 and S^1, and therefore a direction of $\mathrm{conv}(S)$, implying $\bar{p} \in \mathrm{conv}(S)$. Therefore $S^c \subseteq \mathrm{conv}(S)$. ∎

2.2. An Extended Formulation for Q

Consider the following extended formulation of Q

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

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 which are not necessarily polyhedral. Using these definitions, we then show some elementary observations which are known for polyhedral sets.

Definition 1. *Given a closed set $P \subset \mathbb{R}^n$, point $p \in P$ is called an* extreme point *of P if it can not be represented as $p = 1/2 p_1 + 1/2 p_2$ for $p_1, p_2 \in P$, $p_1 \neq p_2$. Set P is called* pointed *if it has extreme points.*

Definition 2. *A closed, pointed set $P \subset \mathbb{R}^n$ 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. Furthermore, 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 also 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 $p = (\bar{w}, \bar{x}) \in \mathrm{conv}(P')$ 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 $\sum_j \lambda_j (ax_j, x_j) = (a\bar{x}, \bar{x}) \in P''$ 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 $\bar{x} \in \mathrm{conv}(P)$, $\bar{x} = \sum_j \lambda_j x_j$ where $x_j \in P$ and $\sum_j \lambda_j = 1$, $\lambda > 0$. In this case, clearly $\sum_j \lambda_j (ax_j, x_j) = (a\bar{x}, \bar{x}) \in \mathrm{conv}(P')$ and therefore $(\bar{w}, \bar{x}) \in \mathrm{conv}(P')$ as $\bar{w} \geq a\bar{x}$. ∎

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

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

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

Proof. Let $D = \{x \in \mathbb{R}^n, y \in \mathbb{R}^n, z \in \times\mathbb{R}^n : (x_i, y_i, z_i) \in S_i,\ \ i = 1, 2, \ldots, n\}$ so that $\bar{Q} = \{w \in \mathbb{R},\ x \in \mathbb{R}^n, y \in \mathbb{R}^n, z \in \times\mathbb{R}^n : w \geq \sum_{i=1}^n q_i y_i,\ (x, y, z) \in D\}$. 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 $\mathrm{conv}(S_i)$ in the description of D. Finally, by Lemma 3, $\bar{Q}^c$ is integral. ∎

2.3. Convex hull description in the original space

Let

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

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

$$\bar{y}_i = \begin{cases} 0 & \bar{z}_i = 0 \\ \bar{x}_i^2/\bar{z}_i & \text{otherwise.} \end{cases}$$

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

$$\|Ax + b\|_2 \leq c^T x + d. \tag{6}$$

The set of points x that satisfy (6) forms a convex set, and efficient and robust algorithms exist for solving optimization problems containing second-order cone constraints [30, 24]. 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

$$x^2 \le yz \text{ with } y \ge 0, z \ge 0, \tag{7}$$

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

$$\|(2x, y-z)^T\| \le y + z.$$

Constraints of the form (7) are often called *rotated second order cone* constraints. The computational benefit of dealing with inequalities (7) 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

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

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

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

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

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

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*

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

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 $(x, 0) \in \text{conv}(W)$ 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 $Q^c(z) = \big\{x \in \mathbb{R}^n \;:\; f_i(x/z) \leq 0$ for $i \in I, \quad uz \geq x \geq lz\big\}$. If all $f_i(x)$ are bounded in $[l, u]$, then,*

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

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. $\text{conv}(W) \;=\; \text{closure}(W^-)$.

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 $\text{conv}(W)$. 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 $q_{it} = \sum_{j=1}^{n} q_{itj}$ and $_i = \max_t\{q_{it}\}$. If all $f_i(x)$ are convex and bounded in $[l, u]$, then $\text{conv}(W) = W^c$, where*

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

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

In this section, we explain how our results are related to earlier works that have appeared in the open 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 [13].

Ceria and Soares [13] 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 $t = 1, \dots, p$, 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 [13] that shows the use of perspective functions to obtain convex hulls of convex sets.

Lemma 9 ([13]). *Let K^t be defined as above for $t \in T = \{1, \dots, T|\}$, and let $K = \mathrm{conv}(\cup_{t=1}^{|T|} K^t)$. Then, $x \in K$ if and only if the following nonlinear system is feasible:*

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

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

Put into this context, 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 Corollary 1 and Lemma 8 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 [13], Frangioni and Gentile [15] introduce the class of *perspective cuts* for mixed integer programs of the form

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

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

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

is valid for the equivalent mixed integer program

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

Frangioni and Gentile [15] 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

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

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

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

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

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

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 [15] are outer approximation cuts for $\mathrm{conv}(P^0 \cup P^1)$. 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 uncapacitated facility location problem, a network design problem with nonlinear congestion constraints, and a portfolio optimization model with buy-in thresholds. 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 ($N = \{1, 2, \ldots, n\}$), a set of facilities ($M = \{1, 2, \ldots, m\}$), and each customer must have its demand for a single commodity met by an open facility. There is a fixed cost c_i for opening a facility $i \in M$. Meeting the demand of customer $j \in N$ from facility $i \in M$ costs an amount proportional to the square of the quantity delivered. A mixed integer nonlinear program for the SQUFL is

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

The variables z_i indicate if facility $i \in N$ is open, and x_{ij} is a decision variable representing the fraction of customer j's demand met from facility i.

To write SQUFL as an indicator-induced MINLP, the auxiliary variables y_{ij} $\forall i \in M, j \in N$ are introduced. The objective function is changed to the linear function

$$\min \sum_{i\in M} c_i z_i + \sum_{i\in M}\sum_{j\in N} q_{ij} y_{ij},$$

and the 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 the indicator variable $z_i = 0$, then $x_{ij} = y_{ij} = 0$ $\forall j \in N$ and the constraints (10) become redundant, while if $z_i = 1$, the constraints (10) become active. Thus, the constraints (10) can be replaced by their perspective counterparts

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

and the resulting relaxation should be significantly tighter.

5.1.1. Computational Results To test the strength of the perspective reformulation, random instances were constructed with facilities and locations uniformly distributed in the unit square. The fixed cost c_i of opening facility $i \in M$ was taken to be randomly and uniformly distributed between 1 and 100. If $p_i \in [0,1]^2$ was the location of facility $i \in M$ and $r_j \in [0,1]^2$ was the location of customer $j \in N$, then the variable cost parameter was calculated as $q_{ij} = 50\|p_i - r_j\|_2$. Günlük et al. [20] constructed instances in a similar manner. For $m \in \{10, 20, 30, 20\}$ and $n \in \{30, 50, 100, 200\}$, ten instances were created and solved using the nonlinear branch-and-bound algorithm available in the open-source MINLP code `BONMIN` [9]. The instances were solved using both the original formulation and the perspective reformulation. All instances were solved on a 1.8GHz AMD Opteron CPU.

Table 1 shows the results of this experiment. In the table, $\bar{z}_R$ represents the average value of the root relaxation of the original formulation, $\bar{z}_P$ the average value of the root relaxation of the perspective reformulation, and $\bar{z}^*$ the average value of the optimal solution found by `BONMIN`. 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. Clearly, reformulating the problem via the perspective reformulation has an enormous impact on the ability to solve the problem. In many cases, the number of nodes in the branch-and-bound tree is orders of magnitude smaller with the perspective reformulation. Also of interest is that nearly all of the integrality gap is closed at the root node by using the perspective reformulation.

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	-	-

The results in Table 1 indicate that the CPU time required to solve one node of the branch-and-bound tree increases dramatically when the perspective formulation is applied. For example, for $n = 30$ and $m = 200$, each node takes on average 383 CPU seconds to evaluate. `BONMIN` uses the interior-point solver `Ipopt` [31] for solving relaxations that arise at nodes of the branch-and-bound tree. `Ipopt` is a solver for general nonlinear programs and is unable to exploit the special second-order cone structure of the inequalities in the perspective reformulation. Furthermore, as the that since the functions $x^2 - yz$ appearing in the perspective reformulation are not convex, `Ipopt` cannot guarantee convergence to a stationary point and its performance is highly dependent on the quality of the initial iterate provided. For the experiments, 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.

To eliminate the obstacles faced by a general NLP solver, the conic formulations were solved with Mosek (version 5) [24], a code specialized for problems of this type. Table 2 shows the number of nodes (N) and CPU seconds (T) required by Mosek v5.0 to solve large random instances of SQUFL formulated with the perspective reformulation wherein the nonlinear inequalities are represented in second-order-cone form. The table also shows the time per node (T/N) when the relaxation is solved by both the SOCP solver and the NLP solver. Note the order-of-magnitude improvement in solution time, which comes solely from the reduced time to solve relaxations at nodes of the branch-and-bound tree. In addition, by using a SOCP solver for the relaxations, larger instances (up to size $n = 50$, $n = 200$) can be solved.

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	-

Günlük et al. [20] derive three classes of cutting planes aimed at strengthening the SQUFL formulation. Table 3, taken from Table 1 of their paper, shows the effectiveness the new cutting planes at reducing the optimality gap at the root node on five instances of different sizes. The perspective reformulation was also performed on these same five instances. In the table, z_R is the value of the root relaxation of the original formulation, z_{GLW} is the value of the root relaxation with three classes of valid inequalities added, z_P is the value of the root relaxation of the perspective reformulation, and z^* is the optimal solution value. The table shows that the perspective reformulation is significantly better at closing the integrality gap than are the cutting planes of Günlük et al. [20].

The largest of the instances in Table 3 was solved to optimality by Lee [22] using `BONMIN`. The solution required 16,697 CPU seconds and 45,901 nodes for the original formulation, and a 21,206 CPU seconds and 29,277 nodes for the

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

formulation with additional inequalities added. The same instance was solved using Mosek on the perspective reformulation wherein the nonlinear inequalities were written as second-order cone constraints. Solution of the instance required only 44 branch-and-bound nodes and *23 CPU seconds* to solve on Intel Pentium 4 CPU with a clock speed of 2.60GHz, a speedup factor of more than 700.

5.2. Network Design with Congestion Constraints

The next application demonstrating the effectiveness of the perspective reformulation is a model for constructing a communication network at minimum cost meeting a design specification for total queuing delay. Similar models appear in the work of Boorstyn and Frank [10], Bertsekas and Gallager [6], and Borchers and Mitchell [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 $\{0\text{-}1\}$ decision 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 the decision variable x_{ij}^k. A typical function to measure the total weighted congestion (or queuing delay) of a flow $f_{ij} = \sum_{k \in K} x_{ij}^k$ in the network 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 the relative importance of the queuing delay that occurs on arc (i, j). 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

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

An observation not previously made in the literature regarding this network design problem is that the congestion inequalities (14) can be written as second-order cone 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

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

Because the inequalities $y_{ij} \ge r_{ij} f_{ij}$ and $u_{ij} \ge f_{ij}$ most hold in any feasible solution, (15) is precisely a constraint in rotated second-order cone form (7).

The *cut-set* inequalities strengthen the relaxation of linear versions of network design problems (without the nonlinear congestion constraints) considerably. The most basic and effective cut-set inequalities simply impose an integral lower bound τ_i on the number of arcs, incident to a given node $i \in N$, that has to be opened. More precisely, 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 $|A'| \le \tau_i - 1$. In this case the associated cut-set inequity for node i is

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

In our computational experiments, we have strengthened the relaxation of the problem by adding these inequalities for both incoming and outgoing arcs for all nodes $i \in N$. More elaborate inequalities could be added (see [8]), but our goal in this work 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. So the formulation can be strengthened using the perspective reformulation. Specifically, each constraint (14) can be replaced by its perspective counterpart:

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

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

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

which is a rotated second-order cone constraint since $y_{ij} \ge r_{ij}f_{ij}$ and $u_{ij}z_{ij} \ge f_{ij}$ 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 To assess the strength of the perspective reformulation of the NDCC, random instances were created. The connectivity of the networks was random, with each potential arc appearing (independently) with probability 0.2. For each instance, the number of commodities was equal to the number of nodes, and each node acted as the unique source for exactly one commodity. For notational purposes, let $s(k)$ be the source node for commodity $k \in K$, and let $B = \sum_{k\in K}\sum_{i\in I\setminus\{s(k)\}} b_i^k$ be the total demand for all commodities. The remainder of the parameters for the instances were created randomly as follows:

$$\begin{aligned}
b_i^k &= \lceil \mathcal{U}(5,25) \rfloor && \forall k \in K\ \forall i \in (I \setminus \{s(k)\}),\\
b_{s(k)}^k &= -\sum_{i\in I\setminus\{s(k)\}} b_i^k && \forall k \in K\\
\\
u_{ij} &= \lceil \mathcal{U}(1.0, 5.0)B/|A| \rfloor && \forall (i,j) \in A\\
r_{ij} &= 1.0 && \forall (i,j) \in A\\
c_{ij} &= \mathcal{U}(1,4) && \forall (i,j) \in A\\
\beta &= \kappa B,
\end{aligned}$$

where $\kappa \in \{1, 2, \ldots\}$ is the smallest integer necessary to make the linear relaxation of the formulation feasible, $\mathcal{U}(a, b)$ is a uniformly distributed random number in the interval (a, b), and $\lceil x \rfloor$ is the closest integer to x.

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. Networks of size $|N| = 20$ and $|N| = 30$ were created. Full tables of results comparing the two formulations can be found in Tables 5 and 6 in the appendix. The computational results show that the perspective reformulation helps the solvability considerably. Of the 35 instances of size $|N| = 20$, 2 can be solved in a four hour 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.

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 [23]. 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):

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

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 $\Delta \in \mathbb{R}^{n \times n}$ [26]. 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 [15], 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 [15] suggest using $D = \lambda_n I$, where $\lambda_n > 0$ is the smallest eigenvalue of Q. In our computational experimentns, 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 [16] 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 $\|w\| = x^T Rx$.

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

$$\begin{aligned}
\text{subject to} \quad w - M^T x &= 0 \\
v - \|w\| &\geq 0 && (19) \\
y_i - x_i^2 &\geq 0 \qquad \forall i \in N && (20) \\
\sum_{i \in N} x_i &= 1 && (21) \\
\sum_{i \in N} \alpha_u x_i &\geq \rho && (22) \\
\ell_i z_i \leq x_i &\leq u_i z_i \qquad \forall i \in N && (23)
\end{aligned}$$

The inequalities (19) can easily be placed in rotated second order cone form (7). 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 (7), so they can be effectively handled by software such as Mosek.

In Table 4 we summarize computational results of an experiment aimed at measuring the effectiveness of the perspective reformulation. In the experiment, twenty instances of the MVOBI problem (ten instances of size $|N| = 200$ and ten instances of size $|N| = 300$) were solved using Mosek with both the original formulation and the perspective reformulation of the constraints (20). The instances were created by Frangioni and Gentile [15], and optimal solutions for the instances are reported at `http://www.di.unipi.it/optimize/Data/MV.html`. Mosek was allowed to run for 10,000 CPU seconds on each instance and formulation. If z_R is the value of the SOCP-relaxation at the root node, z^* is the optimal solution of the problem, z_L and z_U are (respectively) the best lower and upper bounds found by the Mosek branch-and-bound solver, the table reports the root gap to optimal (RGO $= 100(z^* - z_R)/z_R$), the final gap to optimal (FGO $= 100(z^* - z_L)/z_L$), the final gap (FG $= 100(z_U - z_L)/z_L)$), and the number of nodes of the search tree completed in 10,000 seconds.

Table 4. Integrality Gaps of Formulations for MVOBI

	Original Formulation				Perspective Formulation			
Instance	RGO	FGO	FG	Nodes	RGO	FGO	FG	Nodes
P200_a	828.66	244.64	244.64	57287	5.56	2.70	3.71	9838
P200_b	350.71	98.35	106.26	19510	6.60	1.15	0.00	5041
P200_c	344.06	85.46	87.98	23775	6.45	1.70	4.74	6695
P200_d	435.82	120.06	123.73	27395	5.61	1.07	1.35	6015
P200_e	866.25	227.47	233.59	61933	8.12	3.43	3.71	7453
P200_f	820.97	232.60	232.60	45802	6.09	2.47	4.17	8289
P200_g	504.14	130.23	130.23	22253	6.57	2.06	1.56	6999
P200_h	862.87	247.57	257.44	51560	9.38	5.72	8.50	11867
P200_i	808.00	210.97	210.97	62168	6.17	2.92	5.12	8679
P200_j	856.25	221.04	223.73	57111	9.42	6.47	8.88	10308
P300_a	1252.47	507.80	507.80	54394	5.24	4.01	6.17	1968
P300_b	1280.82	552.70	552.70	59721	6.42	4.22	7.75	3211
P300_c	1083.71	449.07	449.29	29049	5.97	3.43	4.50	2828
P300_d	1249.21	527.54	528.89	46800	6.34	4.10	5.69	1980
P300_e	1242.95	515.04	515.04	60751	5.67	3.69	6.29	2164
P300_f	1259.50	513.26	513.74	63967	6.27	4.39	5.50	1946
P300_g	729.05	281.33	282.84	17336	5.05	2.13	5.36	2320
P300_h	1264.86	515.83	517.68	28822	5.80	3.95	4.63	3621
P300_i	1178.57	444.54	451.12	34831	6.52	4.64	7.27	2079
P300_j	1252.22	581.12	581.12	60628	6.26	3.93	6.13	2490

In all cases except one, the Mosek conic IP solver was unable to solve the instance to optimality within the 10,000 second time limit. Nevertheless, the results in the table demonstrate convincingly that the perspective reformulation significantly improves the lower bound. In general, the performance of the conic mixed-integer solver Mosek on the perspective formulation appears inferior to the approach used by Frangioni and Gentile [15] for these instances (see Table 2 in [15].) Their approach is equivalent to linearizing the perspective reformulation at various points, then using the mixed-integer quadratic programming solver CPLEX on the tightened relaxation of the original formulation. The improved performance of a linearization-based approach points 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. Computational results show the power of the proposed techniques—in one case solving instances multiple orders of magnitude faster than reported in the literature. 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 would like to thank Shabbir Ahmed for bringing Proposition 3.3.2 of [5] to their attention. An extended-abstract of this work has been published in the conference proceedings of the Thirteenth Conference on Integer Programming and Combinatorial Optimization (IPCO) [19].

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*, 2008. to appear.
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. A. Borghetti, L. Frangioni, F. Lacalandra, and C. A. Nucci. Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit committment. *IEEE Transacations on Power Systems*, 18:1–10, 2003.
13. S. Ceria and J. Soares. Convex programming for disjunctive optimization. *Mathematical Programming*, 86:595–614, 1999.
14. M. T. Cezik and G. Iyengar. Cuts for mixed 0-1 conic programming. *Mathematical Programming*, 104:179–202, 2005.
15. A. Frangioni and C. Gentile. Perspective cuts for a class of convex 0-1 mixed integer programs. *Mathematical Programming*, 106:225–236, 2006.
16. A. Frangioni and C. Gentile. Sdp diagonalizations and perspective cuts for a class of nonseparable miqp. *Operations Research Letters*, 35:181–185, 2007.
17. R. E. Gomory. An algorithm for the mixed integer problem. Technical Report RM-2597, The RAND Corporation, 1960.
18. I. Grossmann and S. Lee. Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational Optimization and Applications*, pages 83–100, 2003.
19. 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.
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. 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.
22. J. Lee. Mixed-integer nonlinear programming: Some modeling and solution issues. *IBM Journal of Research & Development*, 51:489–497, 2007.
23. H. M. Markowitz. Portfolio selection. *Journal of Finance*, 7:77–91, 1952.
24. mosek. The mosek optimization tools manual. version 5.0 (revision 84), 2008. `www.mosek.com`.
25. G. Nemhauser and L. Wolsey. A recursive procedure for generating all cuts for 0-1 mixed integer programs. *Mathematical Programming*, 46:379–390, 1990.
26. A. F. Perold. Large-scale portfolio optimization. *Management Science*, 30: 1143–1160, 1984.
27. 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.
28. R. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86:515–532, 1999.

29. R. A. Stubbs. *Branch-and-Cut Methods for Mixed 0-1 Convex Programming.* PhD thesis, Northwestern University, December 1996.
30. J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.
31. 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.

7. Appendix

Table 5. Impact of Perspective Reformulation on Network Design Instances. $|N| = |K| = 20$

	Original Formulation					Perspective Formulation				
#	z_{root}	z_L	z_U	N	t	z_{root}	z_L	z_U	N	t
1	102.6	195.5	217.9	28201	14400	213.8	217.9	217.9	816	208
2	83.5	116.0	152.8	44458	14400	147.5	152.8	152.8	6260	1665
3	70.4	112.6	169.9	34679	14400	142.3	161.3	161.3	35454	11901
4	78.9	196.3	243.7	24615	14400	211.4	232.4	232.4	8041	3874
5	77.3	154.0	188.8	35720	14400	168.7	181.7	181.7	2291	662
6	82.4	150.6	189.5	26029	14400	177.2	181.9	181.9	8793	3220
7	73.2	143.8	174.0	36257	14400	154.1	170.6	170.6	6440	1950
8	65.8	105.0	149.0	32688	14400	120.4	140.3	140.3	15757	5400
9	87.3	135.8	167.4	39213	14400	159.4	164.4	164.4	8260	2775
10	77.0	185.1	185.1	4427	2835	173.6	185.1	185.1	530	121
11	76.4	151.8	182.7	36294	14400	158.6	179.4	179.4	6590	2124
12	83.6	126.2	152.5	53915	14400	147.0	152.5	152.5	2538	624
13	83.7	111.9	162.0	53709	14400	137.1	159.1	159.1	3479	691
14	72.1	124.2	186.2	29646	14400	146.4	-1.0	-1.0	-1	66
15	74.5	107.7	142.2	43044	14400	134.9	140.9	140.9	14323	3995
16	75.5	119.7	151.3	41475	14400	127.6	149.4	149.4	11224	3117
17	82.3	106.5	141.5	54256	14400	133.3	140.2	140.2	8990	1723
18	75.4	116.5	191.1	26689	14400	148.8	170.5	170.5	8806	3637
19	88.3	172.1	204.6	32815	14400	189.4	203.3	203.3	1722	468
20	81.9	124.2	175.9	40232	14400	144.4	167.7	167.7	22215	7481
21	88.3	130.0	180.7	22024	14400	168.1	173.3	173.3	5332	2096
22	73.8	157.5	184.0	25888	14400	173.6	181.3	181.3	19585	11498
23	78.5	137.5	216.3	17155	14400	181.8	184.6	190.1	30640	14400
24	75.9	138.9	160.3	41641	14400	140.6	156.9	156.9	1897	550
25	75.8	143.0	176.3	12152	14400	-1.0	-1.0	-1.0	-1	0
26	78.2	209.5	241.0	16793	14400	234.6	238.0	238.0	5584	2969
27	84.0	146.7	195.0	19067	14400	178.1	181.8	181.8	9137	5246
28	90.0	167.8	206.9	38080	14400	184.0	203.9	203.9	22566	8276
29	80.4	141.1	195.3	21225	14400	175.6	182.0	196.7	23303	14400
30	81.9	119.9	166.3	27933	14400	153.3	159.8	159.8	30514	10209
31	101.7	186.2	204.6	14914	14400	197.1	204.2	204.2	1280	350
32	71.9	128.9	172.2	25998	14400	159.6	-1.0	-1.0	-1	14400
33	73.1	117.0	185.2	15279	14400	163.4	-1.0	-1.0	-1	14400
34	78.6	135.8	195.6	25830	14400	154.3	182.6	192.4	28384	14400
35	97.7	229.9	229.9	10678	12043	220.5	229.9	229.9	1019	319

Table 6. Impact of Perspective Reformulation on Network Design Instances. $|N| = |K| = 30$

	Original Formulation					Perspective Formulation				
#	z_{root}	z_L	z_U	N	t	z_{root}	z_L	z_U	N	t
1	167.0	243.6	396.6	3812	14400	375.7	377.8	397.8	5381	14400
2	149.9	248.1	387.4	4211	14400	346.5	348.0	373.0	5931	14400
3	126.3	205.0	355.2	4711	14400	296.3	298.3	329.6	6643	14400
4	137.0	383.8	434.0	3805	14400	425.0	427.5	433.2	5122	14400
5	152.3	235.1	397.5	4304	14400	345.0	346.2	376.1	5634	14400
6	158.5	435.6	479.0	3716	14400	465.0	466.8	477.0	4772	14400
7	172.9	229.6	452.0	3914	14400	374.1	375.2	410.9	4750	14400
8	164.4	343.9	516.4	3305	14400	474.8	476.3	495.8	3512	14400
9	159.6	230.2	381.6	2312	14400	355.2	356.5	375.2	3630	14400
10	137.0	250.6	363.4	3509	14400	340.8	343.2	372.4	5221	14400
11	172.7	244.5	466.8	4513	14400	386.5	389.4	423.1	5660	14400
12	144.6	394.8	446.7	3817	14400	434.3	436.9	442.9	4819	14400
13	141.7	202.0	389.3	3610	14400	326.9	328.2	361.3	4920	14400
14	143.3	294.9	355.7	4724	14400	340.0	344.0	348.1	8848	14400
15	335.5	336.4	358.7	5330	14400	-1.0	-1.0	-1.0	-1	14400
16	144.0	217.0	350.5	4215	14400	328.8	330.9	357.6	4729	14400
17	119.2	180.2	304.3	4408	14400	284.5	286.4	332.3	5522	14400
18	143.3	224.8	360.5	4210	14400	331.2	332.3	362.1	6643	14400
19	126.3	204.9	355.2	4311	14400	296.2	298.9	329.5	6041	14400
20	140.6	201.7	361.2	3607	14400	310.5	311.3	348.0	4434	14400
21	140.3	237.0	366.4	4208	14400	332.4	335.0	357.9	6043	14400
22	131.2	181.6	269.7	6708	14400	248.3	254.6	271.3	11942	14400
23	128.1	230.2	360.7	4507	14400	311.9	314.3	336.6	7962	14400
24	167.3	464.6	522.9	2915	14400	512.0	512.6	522.9	3549	14400
25	154.3	215.3	354.0	3013	14400	338.1	339.7	363.9	3227	14400
26	130.7	198.9	341.6	2407	14400	309.9	311.2	332.0	3632	14400
27	138.7	406.3	428.6	3160	14400	423.4	425.7	426.5	3647	14400
28	130.5	197.2	392.5	3614	14400	307.0	309.1	342.6	4737	14400
29	150.0	203.9	433.2	4412	14400	337.6	338.9	366.6	4728	14400
30	139.6	274.4	349.1	3714	14400	331.0	333.6	344.2	6059	14400
31	137.0	236.4	349.9	3511	14400	322.9	325.2	345.3	6633	14400
32	162.6	313.3	412.1	3208	14400	376.1	379.4	389.2	4616	14400
33	140.1	205.2	433.9	2907	14400	320.5	322.8	367.5	3822	14400
34	152.4	274.6	424.5	2405	14400	397.3	397.8	417.3	4526	14400
35	135.4	199.4	376.3	4106	14400	318.5	319.8	364.3	5442	14400