Anureet Saxena · Pierre Bonami · Jon Lee

Convex Relaxations of Non-Convex Mixed Integer Quadratically Constrained Programs: Projected Formulations

November 19, 2008

Abstract A common way to produce a convex relaxation of a Mixed Integer Quadratically Constrained Program (MIQCP) is to lift the problem into a higher dimensional space by introducing variables Y_{ij} to represent each of the products $x_i x_j$ of variables appearing in a quadratic form. One advantage of such extended relaxations is that they can be efficiently strengthened by using the (convex) SDP constraint $Y - xx^T \succeq 0$ and disjunctive programming. On the other hand, their main drawback is their huge size, even for problems of moderate size. In this paper, we study methods to build low-dimensional relaxations of MIQCP that capture the strength of the extended formulations. To do so, we use projection techniques pioneered in the context of the lift-and-project methodology. We show how the extended formulation can be algorithmically projected to the original space by solving linear programs. Furthermore, we extend the technique to project the SDP relaxation by solving SDPs. In the case of an MIQCP with a single quadratic constraint, we propose a subgradient-based heuristic to efficiently solve these SDPs. We also propose a new *eigen reformulation* for MIQCP, and a cut generation technique to strengthen this reformulation using polarity. We present extensive computational results to illustrate the efficiency of the proposed techniques. Our computational results have two highlights. First, on the GLOBALLib instances, we are able to generate relaxations that are almost as strong as those proposed in our companion paper even though our computing times are about 100 times smaller, on average. Second, on the box QP instances, the strengthened relaxations generated by our code are almost as strong as the well-studied SDP+RLT relaxations and can be solved in less than 2 sec even for larger instances with 100 variables; the SDP+RLT relaxations of the same set of instances can take up to a couple of hours to solve using a state-of-the-art SDP solver.

Anureet Saxena
Axioma Inc., 8800 Roswell Road, Building B. Suite 295, Atlanta GA, 30350.
E-mail: asaxena@axiomainc.com

Pierre Bonami
Laboratoire d'Informatique Fondamentale de Marseille,
CNRS-Marseille Universités, France
E-mail: pierre.bonami@lif.univ-mrs.fr

Jon Lee
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 USA
E-mail: jonlee@us.ibm.com

1 Introduction

In this paper we study the mixed integer quadratically constrained program defined as follows:

$$(\textbf{MIQCP}) \qquad \begin{array}{l} \min\ a_0^T x \\ \text{s.t.} \\ x^T A_k x + a_k^T x + b_k \leq 0\ , \quad k = 1 \ldots m\ ; \\ x_j \in \mathbb{Z}\ , \quad j \in N_I\ ; \\ l \leq x \leq u\ , \end{array}$$

where $N = \{1, \ldots, n\}$ denotes the set of variables, $N_I = \{1, \ldots, p\}$ denotes the set of integer constrained variables, $M = \{1, \ldots, m\}$ denotes the index set of constraints, A_k $(k = 1, \ldots, m)$ are $n \times n$ symmetric (usually not positive semidefinite) matrices, a_k $(k = 0, \ldots, m)$, l and u are n-dimensional vectors and b_k $(k = 1, \ldots, m)$ are scalars. For ease of exposition, we assume that all variables have finite lower and upper bounds. All results presented in this paper can be generalized easily to the case in which only variables appearing in bilinear terms are assumed to have finite bounds. **MIQCP**s arise in a wide range of practical applications such as chemical process design, optimal control problems, combinatorial optimization etc. Furthermore, any polynomial programming problem can be transformed into a **MIQCP** by introducing additional variables making **MIQCP** a fairly versatile optimization model.

From a computational standpoint, **MIQCP**s can be very difficult to solve in practice because they combine two kinds of non-convexities, namely, integer variables and non-convex quadratic constraints. One of the standard approaches for solving **MIQCP** entails introducing additional variables $Y_{ij} = x_i x_j$ representing bilinear terms, and then working in the extended space of (x, Y) variables. The resulting relaxation can be strengthened by adding the so-called RLT inequalities and the positive semidefiniteness condition $Y - xx^T \succcurlyeq 0$; we refer to the strengthened relaxation of **MIQCP** obtained in this manner as **MIQCP-SDP** in the sequel. **MIQCP-SDP** has been extensively studied in the past one decade and good progress, both theoretical and computational, has been made. In our companion paper [17] we investigate further strengthening of **MIQCP-SDP** relaxation using disjunctive cuts and report promising computational results. Despite these early successes, the presence of an enormous number $O(n^2)$ of additional Y_{ij} variables continues to haunt researchers pursuing this line of research. This problem gets even more aggravated for branch-and-bound algorithms which have to carry the burden of these large relaxations at every node of the enumeration tree. Naturally we are interested in relaxations that capture the strength of these extended formulations but are defined only in the space of x variables. Systematic theoretical and computational investigation of such projected relaxations constitutes the topic of this paper.

We employ the lift-and-project methodology developed and nurtured by Balas [2,3] over the last three decades as the workhorse in our enterprise. There are three main ingredients to the results presented in this paper, namely, projection cones, surrogate constraints and convexification schemes. Projection cones were introduced by Balas [3] to characterize the class of undominated inequalities that arise in projection of polyhedral sets. When intersected with a normalization constraint, these projection cones readily yield a linear-programming based separation algorithm for the projected inequalities; our main cut generator **ProjLP** is derived from such a linear program. We extend the reasoning of Balas to derive a constructive characterization of the projection of the **MIQCP-SDP** relaxation. The separation program in this case turns out to be a semidefinite program (SDP). We show that the separation SDP can be cast as a piecewise-linear convex optimization problem over the Cartesian product of the cone of positive semidefinite matrices and the simplex. A projected subgradient heuristic to solve the resulting convex program for the special case in which there is only one constraint (i.e $m = 1$) is briefly discussed; our computational results demonstrate that the proposed heuristic has promising practical performance.

The concept of a surrogate constraint, a constraint obtained by taking non-negative combination of other problem constraints, has played a pivotal role in mixed integer linear programming (MILP). Separation routines for various classes of MILP cutting planes such as mixed integer Gomory cuts, intersection cuts, mixed integer rounding cuts, various kinds of cover cuts, etc., usually involve aggregating original constraints to create a surrogate constraint which is subsequently used to derive cuts. Although a well-studied concept in MILP literature, a systematic study of surrogate constraints in the context of **MIQCP** has remained an unchartered territory. We discuss techniques for detecting useful surrogate constraints by utilizing the optimal solutions to our separation programs.

Convexification of a non-convex constraint, say $x^T Ax + a^T x + b \leq 0$, obtained by replacing the constituent bilinear terms $A_{ij}x_ix_j$ by their McCormick estimators [13,19] has been extensively studied in the literature. In this paper, we study an alternative convexification scheme that splits the Hessian matrix A as a difference of a positive semidefinite (PSD) matrix B and a symmetric matrix C, i.e. $A = B - C$; while the PSD matrix B is used to derive a convex term $x^T Bx$ in the cut, its non-convex alter ego $x^T Cx$ is convexified by replacing $C_{ij}x_ix_j$ terms by their McCormick estimators. As our results show, a systematic application of this convexification scheme over all possible surrogate constraints yields the projection of the **MIQCP-SDP** relaxation to the space of x variables.

We introduce an alternative reformulation of **MIQCP**, referred to as *eigen reformulation*, which identifies directions of maximal non-convexity in each constraint and introduces additional variables to expose them. We propose a cut-generation scheme that works with the projection of **MIQCP** along a subset of these directions, computes the extreme points of the projection and embeds the extreme points within the polarity framework to derive polarity cuts. By virtue of additional problem constraints, the geometry of **MIQCP** along these directions of maximal non-convexity tend to be highly correlated, and our cut generator identifies and exploits these correlations to generate strong cutting planes for **MIQCP**. The idea of eigen reformulation has some similarities to the work of Kim and Kojima [10] although its treatment in our paper is more central.

We demonstrate the computational value of our results through a series of experiments. These experiments were conducted on a test bed comprising instances from GLOBALLib [9], instances from Lee and Grossmann [11], and Box-QP instances from [22]. Besides reporting the strengths of various relaxations examined in this paper, we also study the marginal impact of various classes of cutting planes and compare our results with those presented in our companion paper [17]. Our computational results have two highlights. First, on the GLOBALLib instances we are able to generate relaxations that are almost as strong as those proposed in [17] even though our computing times are about 100 times smaller, on average. Second, on the box QP instances the strengthened relaxations generated by our code are almost as strong as the **MIQCP-SDP** relaxation and can be solved in less than 2 *sec* even for larger instances with 100 variables; the **MIQCP-SDP** relaxations of the same set of instances can take up to a couple of hours to solve using a state-of-the-art SDP solver.

The rest of the paper is organized as follows. Sections 2 and 3 discuss the projection of the extended RLT and SDP relaxation of **MIQCP**, respectively. In section 4 we develop the notion of eigen reformulation and discuss systematic techniques for deriving strong valid cutting planes for **MIQCP** by computing low dimensional projections of its relaxations. We present our computational results in section 5 and conclude with remarks on generalizations to non-convex MINLPs in section 6.

2 Projecting the Extended RLT Formulation

A standard approach to derive a convex relaxation of **MIQCP** is to introduce additional variables $Y_{ij} = x_ix_j$ and replace the quadratic constraints by $A_k.Y + a_k^T x + b_k \leq 0$ $(k \in M)$[1]. The resulting formulation can be strengthened by adding the so-called RLT inequalities [13,19] to yield the following lifted relaxation of **MIQCP**.

(**MIQCP-RLT**)

$$\begin{aligned} &\min\ a_0^T x \\ &\text{s.t.} \\ &A_k.Y + a_k^T x + b_k \leq 0\ , \quad k \in M\ ; \\ &l_j \leq x_j \leq u_j\ , \quad \forall j \in N\ ; \\ &y_{ij}^-(x) \leq Y_{ij} \leq y_{ij}^+(x)\ , \quad \forall i,j \in N\ , \end{aligned}$$

where

$$\begin{aligned} y_{ij}^-(x) &= \max\ \{u_ix_j + u_jx_i - u_iu_j, l_ix_j + l_jx_i - l_il_j\} \quad \forall i,j\ , \\ y_{ij}^+(x) &= \min\ \{l_ix_j + u_jx_i - l_iu_j, u_ix_j + l_jx_i - u_il_j\} \quad \forall i,j\ . \end{aligned}$$

[1] For symmetric matrices A and B of conformable dimensions, we define $A.B = tr(AB)$.

Let $P_{(x,Y)}$ denote the set of feasible solutions to **MIQCP-RLT**, and let $Q_x = \{x \in \mathbb{R}^N \mid \exists Y \text{ s.t. } (x,Y) \in P_{(x,Y)}\}$ denote the projection of $P_{(x,Y)}$ to the space of x-variables. The theorem that follows gives a constructive characterization of the projection Q_x. The proof of the theorem follows immediately from results of Balas [3].

Theorem 1 *Suppose that $\hat{x} \in \mathbb{R}^N$ satisfies $l_j \leq \hat{x}_j \leq u_j \ \forall j$. Then $\hat{x} \in Q_x$ if and only if the optimal value of the following linear program is non-positive.*

(**ProjLP**)

$$\begin{aligned}
&\max \ \textstyle\sum_{i,j} \left(B_{ij} y^-_{ij}(\hat{x}) - C_{ij} y^+_{ij}(\hat{x})\right) + \sum_{k\in M} u_k \left(a_k^T \hat{x} + b_k\right) \\
&\text{s.t.} \\
&\textstyle\sum_{k\in M} u_k A_k - B + C = 0 \ ; \\
&\textstyle\sum_{k\in M} u_k = 1 \ ; \\
&u_k \geq 0, \quad \forall k \in M \ ; \\
&B_{ij}, C_{ij} \geq 0, \quad \forall i,j \in N \ .
\end{aligned}$$

Furthermore, if (u, B, C) is a feasible solution to **ProjLP** *having positive objective value, then*

$$\sum_{i,j} \left(B_{ij} y^-_{ij}(x) - C_{ij} y^+_{ij}(x)\right) + \sum_{k\in M} u_k \left(a_k^T x + b_k\right) \leq 0 \tag{1}$$

is a valid convex inequality for Q_x that cuts off $\hat{x}$.

The constraint $\sum_{k\in M} u_k = 1$ in **ProjLP** is a normalization constraint that, along with the hypothesis $l_j \leq \hat{x}_j \leq u_j \ \forall j$, guarantees the boundedness of **ProjLP**. Consider the dual of **ProjLP** :

(**DProjLP**)

$$\begin{aligned}
&\min \ \eta \\
&\text{s.t.} \\
&-A_k.Y + \eta \geq a_k^T \hat{x} + b_k \ , \quad \forall k \in M \ ; \\
&y^-_{ij}(\hat{x}) \leq Y_{ij} \leq y^+_{ij}(\hat{x}) \ , \quad \forall i,j \in N \ .
\end{aligned}$$

DProjLP is a linear program with m constraints and n^2 variables; note that $y^-_{ij}(\hat{x}) \leq Y_{ij} \leq y^+_{ij}(\hat{x})$ can be handled as (simple) bound constraints on the Y_{ij} variables. Typically $m << n^2$, and hence from a computational standpoint it is much more efficient to solve **DProjLP** than **ProjLP**. Furthermore, if $A_k = 0$ and $a_k^T \hat{x} + b_k \leq 0$, then the corresponding constraint can be dropped from **DProjLP**. Alternatively, the number of non-trivial (i.e non-bound type) constraints in **DProjLP** is exactly equal to the number of quadratic constraints in **MIQCP**. In our computational experiments, we solved **DProjLP** and used the optimal dual solution associated with **DProjLP** to derive the projected inequality. Several remarks are in order.

First, **DProjLP** handles the enormous number $O(n^2)$ of RLT inequalities as bounds on the Y_{ij} variables. Because computationally intensive components of most linear programming algorithms (basis update in simplex-type algorithms and solving linear systems in Interior Point Methods (IPM)) depend only on the number of non-trivial constraints, this feature of **DProjLP** significantly reduces the computational overheads associated with using RLT inequalities.

Second, **DProjLP** can be solved by either a simplex-type algorithm or by IPM. Preliminary computational experiments suggest that IPM have an upper hand over simplex-type methods in solving **DProjLP**. We suspect two reasons for this behavior. First, the bounds on the Y_{ij} variables change radically from one iteration to the next thereby diminishing the warm-start capabilities of simplex-type procedures. Second, IPM used without a crossover phase tend to converge faster to the optimal solution than simplex-type algorithms. In our computational experiments, we used the Barrier Algorithm in CPLEX 10.1 to solve **DProjLP**.

Third, Theorem 1 can be easily modified to handle convex quadratic constraints of the form $A.Y + x^T Dx + a^T x + b \leq 0$, with $D \not\succeq 0$; the modification entails using $\left(\hat{x}^T D\hat{x} + a^T \hat{x} + b\right)$ instead of $\left(a^T \hat{x} + b\right)$ in the objective function of **ProjLP**. Such convex quadratic cuts might arise, for instance, while strengthening the extended formulation of **MIQCP** (see [16,17]).

Fourth, if the optimal value of **DProjLP**, or equivalently **ProjLP**, is non-positive and $(\hat{Y}, \hat{\eta})$ is an optimal solution of **DProjLP**, then $(\hat{x}, \hat{Y}) \in P_{(x,Y)}$. In other words, $\hat{Y}$ provides a certificate of containment for $\hat{x}$ (i.e., a certificate that $\hat{x} \in Q_x$).

Fifth, if (u, B, C) is a feasible solution to **ProjLP** then the convex inequality

$$\sum_{i,j} \left(B_{ij} y_{ij}^{-}(x) - C_{ij} y_{ij}^{+}(x)\right) + \sum_{k \in M} u_k \left(a_k^T x + b_k\right) \leq 0$$

is equivalent to an exponentially large set of linear inequalities obtained by replacing the $y_{ij}^{-}(x)$ and $y_{ij}^{+}(x)$ terms by one of the linear expressions used in defining them. It is easy to show that there exists a straightforward linear time separation algorithm for this exponentially large set of linear inequalities. In our computational experiments, we used only the most violated linear inequality among all of them.

We conclude this section by giving a reformulation of **ProjLP** which gives some additional insights into the geometry of this linear program. For $x \in \mathbb{R}$, we define $x^+ = \max\{x, 0\}$ and $x^- = \min\{x, 0\}$. Also, we denote the standard simplex in $\mathbb{R}^M$ by

$$\Sigma_M = \left\{ u \in \mathbb{R}^M \mid \sum_{k \in M} u_k = 1,\ u \geq 0 \right\}.$$

Theorem 2 *Suppose that $\hat{x} \in \mathbb{R}^N$ satisfies $l_j \leq \hat{x}_j \leq u_j\ \forall j$. Then* **ProjLP** *is equivalent to the following convex piecewise linear optimization problem over the standard simplex Σ_M.*

(RLT-ProjCP) $$\max \{F(u) \mid u \in \Sigma_M\},$$

where

$$\begin{aligned} F(u) = & \sum_{i,j} \left(\sum_{k \in M} u_k A_{ij}^k \right)^+ \left(y_{ij}^{-}(\hat{x}) - \hat{x}_i \hat{x}_j\right) \\ & + \sum_{i,j} \left(\sum_{k \in M} u_k A_{ij}^k \right)^- \left(y_{ij}^{+}(\hat{x}) - \hat{x}_i \hat{x}_j\right) \\ & + \sum_{k \in M} u_k (\hat{x}^T A_k \hat{x}) + \sum_{k \in M} u_k \left(a_k^T \hat{x} + b_k\right) \ . \end{aligned}$$

Proof The concavity of $F(u)$ follows immmediately from the observation $y_{ij}^{-}(\hat{x}) \leq \hat{x}_i \hat{x}_j \leq y_{ij}^{+}(\hat{x})$. Because **RLT-ProjCP** entails maximizing a concave function over a convex domain, it is a convex optimization problem. Suppose $u \in \Sigma_M$, and let $B_{ij} = \left(\sum_{k \in M} u_k A_{ij}^k\right)^+$ and $C_{ij} = -\left(\sum_{k \in M} u_k A_{ij}^k\right)^-\ \forall i, j$. Clearly, (u, B, C) is a feasible solution to **ProjLP**, $\sum_{k \in M} u_k A_k = B - C$, and

$$F(u) = \sum_{i,j} \left(B_{ij} y_{ij}^{-}(\hat{x}) - C_{ij} y_{ij}^{+}(\hat{x})\right) + \sum_{k \in M} u_k \left(a_k^T \hat{x} + b_k\right)$$

, which implies that the optimal objective value of **ProjLP** is no more than the optimal objective value of **RLT-ProjCP**. Conversely, suppose that (u, B, C) is an optimal solution to **ProjLP**. Because $l_j \leq \hat{x}_j \leq u_j\ \forall j$, then $y_{ij}^{-}(\hat{x}) \leq y_{ij}^{+}(\hat{x})$, which implies that $B_{ij} C_{ij} = 0\ \forall i, j$. Hence, $B_{ij} = \left(\sum_{k \in M} u_k A_{ij}^k\right)^+$ and $C_{ij} = -\left(\sum_{k \in M} u_k A_{ij}^k\right)^-\ \forall i, j$, and the objective function value of (u, B, C) w.r.t. **ProjLP** is exactly equal to $F(u)$. □

Theorem 2 shows that **ProjLP** is essentially an unconstrained optimization problem. This is not surprising as the process of deriving a projected inequality via **ProjLP** has the following simple two-phase interpretation. The first phase uses the u_k $(k \in M)$ multipliers to take a non-negative combination of the constraints and derive a surrogate constraint of the form $x^T A x + a^T x + b \leq 0$. The second phase splits the Hessian matrix A as a difference of two non-negative matrices, say $A = B - C$, with $B, C \geq 0$, and then uses $B_{ij} y_{ij}^-(x)$ and $C_{ij} y_{ij}^+(x)$ to approximate the binomial terms $B_{ij} x_i x_j$ and $C_{ij} x_i x_j$, respectively. Formally,

$$x^T A x + a^T x + b \leq 0 \ ,$$
$$x^T B x - x^T C x + a^T x + b \leq 0 \ ,$$
$$\sum_{i,j} \left(B_{ij} y_{ij}^-(x) - C_{ij} y_{ij}^+(x) \right) + a^T x + b \leq 0 \ \left(\text{because } y_{ij}^-(x) \leq x_i x_j \leq y_{ij}^+(x) \right) \ .$$

Note that the above two-phase procedure can be carried out for any set of non-negative multipliers u. Furthermore, once the surrogate constraint has been obtained, we can use the following alternative splitting of the Hessian matrix, $A = B + C - D$, with $B \succcurlyeq 0, \ C, D \geq 0$, and derive the following valid convex quadratic inequality for **MIQCP**:

$$x^T B x + a^T x + b + \sum_{i,j} \left(C_{ij} y_{ij}^-(x) - D_{ij} y_{ij}^+(x) \right) \leq 0 \ .$$

As the results of the following section show, the relaxation of **MIQCP** obtained by adding all such convex quadratic cuts is identical to the projection of the SDP relaxation $P_{(x,Y)} \cap \{(x, Y) \mid Y - xx^T \succcurlyeq 0\}$ of **MIQCP** to the space of x-variables.

3 Projecting the SDP Formulation

Note that **MIQCP-RLT** can be strengthened by adding the convex constraint $Y - xx^T \succcurlyeq 0$; the resulting strengthened relaxation is referred to as **MIQCP-SDP** in the sequel. Let $P_{(x,Y)}^+ = P_{(x,Y)} \cap \{(x, Y) \mid Y - xx^T \succcurlyeq 0\}$ denote the set of feasible solutions of the resulting strengthened formulation. Similarly, let $Q_x^+ = \{x \mid \exists Y \text{ s.t. } (x, Y) \in P_{(x,Y)}^+\}$ denote the projection of $P_{(x,Y)}^+$ to the space of x variables. The theorem that follows gives a constructive characterization of Q_x^+.

Theorem 3 *Suppose that $\hat{x} \in \mathbb{R}^N$ satisfies $l_j \leq \hat{x}_j \leq u_j \ \forall j$. Then $\hat{x} \in Q_x^+$ if and only if the optimal value of the following semidefinite program (SDP) is non-positive.*

(ProjSDP)

$$\begin{aligned}
&\max \ \textstyle\sum_{i,j} B.\hat{x}\hat{x}^T + \left(C_{ij} y_{ij}^-(\hat{x}) - D_{ij} y_{ij}^+(\hat{x}) \right) + \sum_{k \in M} u_k \left(a_k^T \hat{x} + b_k \right) \\
&\text{s.t.} \\
&\textstyle\sum_{k \in M} u_k A_k - B - C + D = 0 \ ; \\
&\textstyle\sum_{k \in M} u_k = 1 \ ; \\
&u_k \geq 0, \quad \forall k \in M \ ; \\
&C_{ij}, D_{ij} \geq 0, \quad \forall i, j \in N \ ; \\
&B \succcurlyeq 0 \ .
\end{aligned}$$

Furthermore, if (u, B, C, D) is a feasible solution to **ProjSDP** *with positive objective value, then*

$$x^T B x + \sum_{i,j} \left(C_{ij} y_{ij}^-(x) - D_{ij} y_{ij}^+(x) \right) + \sum_{k \in M} u_k \left(a_k^T x + b_k \right) \leq 0 \tag{2}$$

is a valid convex inequality for Q_x^+ that cuts off $\hat{x}$.

Proof Let **ProjSDP'** denote the SDP obtained from **ProjSDP** by replacing the normalization constraint $\sum_{k \in M} u_k = 1$ by $\sum_{k \in M} u_k + tr(B) = 1$. Note that if (u, B, C, D) is a feasible solution to **ProjSDP'** with positive objective value then $\sum_{k \in M} u_k > 0$. This implies that **ProjSDP** has a positive objective value if and only if **ProjSDP'** has a positive objective value. Consider the dual of **ProjSDP'** :

(DProjSDP')

$$\begin{aligned}
&\min \ \eta \\
&\text{s.t.} \\
&-A_k.Y + \eta \geq a_k^T \hat{x} + b_k, \quad \forall k \in M \ ; \\
&Y + \eta I - \hat{x}\hat{x}^T \succcurlyeq 0 \ ; \\
&y_{ij}^-(\hat{x}) \leq Y_{ij} \leq y_{ij}^+(\hat{x}), \quad \forall i, j \in N \ .
\end{aligned}$$

Clearly, **DProjSDP'** is a strictly feasible SDP, the optimal value of which is non-positive if and only if $\hat{x} \in Q_x^+$. The result follows from SDP duality in the presence of the weak Slater qualification condition. □

Note that unlike **ProjLP**, the separation program **ProjDSP** of Theorem 3 is a semidefinite program. This observation has far reaching consequences due to the differences in technology available to solve linear programs and semidefinite programs, their membership in the class of polynomial time solvable problems notwithstanding. For instance, **ProjLP** arising from problems with 100 variables can be solved in a fraction of a second, whereas **ProjSDP** for the same instance may take up to a couple of minutes to solve. Furthermore, while the presence of RLT inequalities as bound constraints $y_{ij}^-(\hat{x}) \leq Y_{ij} \leq y_{ij}^+(\hat{x})$ significantly speeds up the linear programming algorithm used to solve **ProjLP**, the same set of constraints renders **ProjSDP** tremendously more difficult to solve due to the inability of current SDP solvers to efficiently handle non-trivial bound constraints on matrix entries. Preliminary experimentation with black-box SDP solvers clearly indicates the practical limitations of using **ProjSDP** directly.

Similar to Theorem 2, the theorem that follows gives an alternative reformulation of **ProjSDP**. The proof of the theorem is similar to that of Theorem 2.

Theorem 4 *Suppose that $\hat{x} \in \mathbb{R}^N$ satisfies $l_j \leq \hat{x}_j \leq u_j \ \forall j$. Then* **ProjSDP** *is equivalent to the following convex piecewise linear optimization problem over the Cartesian product of the cone of positive semidefinite matrices and the simplex,*

(SDP-ProjCP) $$\max\left\{F(u) \mid u \in \Sigma_M \ , \ B \succcurlyeq 0\right\},$$

where

$$\begin{aligned} F(u,B) = &\sum_{i,j}\left(\sum_{k\in M} u_k A_{ij}^k - B_{ij}\right)^+ \left(y_{ij}^-(\hat{x}) - \hat{x}_i\hat{x}_j\right) \\ &+ \sum_{i,j}\left(\sum_{k\in M} u_k A_{ij}^k - B_{ij}\right)^- \left(y_{ij}^+(\hat{x}) - \hat{x}_i\hat{x}_j\right) \\ &+ \sum_{k\in M} u_k(\hat{x}^T A_k \hat{x}) + \sum_{k\in M} u_k\left(a_k^T\hat{x} + b_k\right) \ . \end{aligned}$$

Furthermore, if $u \in \Sigma_M$ and $B \succcurlyeq 0$ satisfy $F(u,B) > 0$, then

(**SDP-Cut**)
$$\begin{aligned} &\textstyle\sum_{i,j}\left(\sum_{k\in M} u_k A_{ij}^k - B_{ij}\right)^+ \left(y_{ij}^-(x) - x_i x_j\right) \\ &+ \textstyle\sum_{i,j}\left(\sum_{k\in M} u_k A_{ij}^k - B_{ij}\right)^- \left(y_{ij}^+(x) - x_i x_j\right) \\ &+ \textstyle\sum_{k\in M} u_k\left(x^T A_k x\right) + \sum_{k\in M} u_k\left(a_k^T x + b_k\right) \leq 0 \end{aligned}$$

is a valid convex inequality for Q_x^+ that cuts off $\hat{x}$.

The above theorem suggests that **ProjSDP** can probably be solved more efficiently by applying a subgradient algorithm to **SDP-ProjCP**. A detailed theoretical and computational investigation of such an algorithm requires a good understanding of interior point methods and goes beyond the scope of the current paper. Nevertheless, for the purpose of illustration, we designed the following heuristic to solve **SDP-ProjCP** for the special case when $|M| = 1$. The heuristic computes a subgradient of $F(u,B)$ with respect to B in each iteration, projects the subgradient to the cone of positive semidefinite matrices and performs line-search along the resulting direction. While the heuristic is not guaranteed to yield a provably optimal solution, it seems to have good practical performance (see Section 5 for computational results on the Box-QP instances).

Projected Subgradient Heuristic for SDP-ProjCP

Input A non-convex constraint $x^T A x + a^T x + b \leq 0$, a positive semidefinite matrix B, and the incumbent solution $\hat{x}$ that we seek to cut off. [2]

Algorithm

[2] Because $\Sigma_M = \{(1)\}$ for the special case when $|M| = 1$, we drop u in $F(u,B)$ to simplify the notation.

1. Compute a subgradient $\bar{B}$ of the piecewise-linear function F at B.
2. Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of $\bar{B}$, and let $v_1, \ldots, v_n$ denote the associated eigenvectors.
3. Let $B^+ = \sum_{\lambda_k > 0} \lambda_k v_k v_k^T$ denote the projection of $\bar{B}$ onto the cone of positive semidefinite matrices.
4. Solve the one-dimensional convex optimization problem

$$\max_{\theta \geq 0} \; F(B + \theta B^+) \; , \tag{3}$$

and let $\bar{\theta}$ denote the optimal solution.
5. If $F(B + \bar{\theta} B^+) > F(B)$ then set $B := B + \bar{\theta} B^+$ and goto step 1.
6. If $F(B) > 0$ then generate **SDP-Cut**.
7. Stop.

In order to improve numerical and empirical behavior of the above heuristic, we made the following two modifications. First, we imposed an iteration limit of 200 to avoid infinite loops as well as a long sequence of potentially small improvements. Second, instead of solving the one-dimensional problem (3) to optimality, we solve it only approximately by performing at most K iterations of a standard bisection-search algorithm; we chose $K = 5$ in our implementation.

Note that the B matrix always remains positive semidefinite and hence feasible to **SDP-ProjCP** in each iteration of the above algorithm. We used the spectral decomposition of the Hessian matrix A to initialize the B matrix. In particular, if $\mu_1, \ldots, \mu_n$ are the eigenvalues of A and $v_1, \ldots, v_n$ are the associated eigenvectors, we initialize $B = \sum_{\mu_k > 0} \mu_k v_k v_k^T$.

Theorems 1 and 3 propose convexification techniques which are applied to surrogate constraints obtained by taking non-negative combination of the original constraints. Theorem 1 proposes a 2-way splitting of the Hessian matrix, say $A = B - C$ $(B, C \geq 0)$, and approximating the atomic non-convex expressions $B_{ij} x_i x_j$ and $C_{ij} x_i x_j$ by their convex estimators $B_{ij} y_{ij}^-(x)$ and $C_{ij} y_{ij}^+(x)$, respectively. Theorem 3, on the other hand, proposes a 3-way splitting of the Hessian matrix to derive (2). Both of these theorems have a common lacunae, namely, that neither of them exploits additional problem constraints during the convexification process except during the construction of the surrogate constraint. As the following example shows, there is a lot to be gained by engaging these additional constraints in the convexification process.

Consider the **MIQCP** shown below,

$$\begin{array}{l} \min \; x_3 \\ \text{s.t.} \\ x_1 x_2 - x_1 - x_2 - x_3 \leq 0 \\ -6x_1 + 8x_2 \leq 3 \\ 3x_1 - x_2 \leq 3 \\ 0 \leq x_1, x_2 \leq 1.5 \; . \end{array}$$

The above **MIQCP** was derived from the st_e23 instance in the GLOBALLib [9] repository by strengthening bounds on x_1 and x_2. Suppose $\hat{x} = (0.81107, 0.68893, -1.5)$ is the incumbent solution which we want to cut off. Because the above **MIQCP** has a single non-linear constraint, the unique surrogate constraint examined by Theorems 1 and 3 is given by $x_1 x_2 - x_1 - x_2 - x_3 \leq 0$. Let $P_1 = \mathit{clconv}\,\{x \mid x_1 x_2 - x_1 - x_2 - x_3 \leq 0, \; 0 \leq x_1, x_2 \leq 1.5\}$. Note that $(1.5, 0, -1.5) \in P_1$, $(0, 1.5, -1.5) \in P_1$ and

$$\begin{array}{lll} 0.5407 \; (1.5, 0, -1.5) & + \; 0.4593 \; (0, 1.5, -1.5) & = \hat{x} \; , \\ 0.5407 & + \; 0.4593 & = 1 \; , \end{array}$$

which implies that $\hat{x} \in P_1$. Consequently, any cut generator which uses only the surrogate constraint and bounds information cannot cut off $\hat{x}$. In particular, $\hat{x}$ cannot be cut off by inequalities (1) and (2).

Next consider the following reformulation of the surrogate constraint obtained by using the spectral decomposition of its Hessian matrix $\begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix}$,

$$\frac{1}{2}\left(x_1 + x_2\right)^2 - x_1 - x_2 - x_3 \leq \frac{1}{2}\left(x_1 - x_2\right)^2 \; .$$

Note that additional problem constraints, namely $-6x_1 + 8x_2 \leq 3$ and $3x_1 - x_2 \leq 3$ can be used to derive lower and upper bounds on the linear function $x_1 - x_2$ over the feasible region; by solving a pair of linear programs, we determined these bounds to be $-0.375 \le x_1 - x_2 \le 1$. These bounds, in turn, can be used to approximate $(x_1 - x_2)^2$ by its secant approximation $0.625(x_1 - x_2) + 0.375$ on the $[-0.375, 1]$ interval and derive the cut

$$\frac{1}{2}(x_1 + x_2)^2 - x_1 - x_2 - x_3 \le \frac{1}{2}(0.625(x_1 - x_2) + 0.375) \ ,$$

which cuts off $\hat{x}$. In the next section, we develop this idea and embed it within the polarity framework to derive cutting planes for **MIQCP**.

4 Low Dimensional Projections

In this section we describe a systematic technique for deriving strong valid cutting planes for **MIQCP** by computing low dimensional projections of its relaxations.

Suppose $x^T Ax + a^T x + b \le 0$ is a quadratic inequality that is satisfied by all feasible solutions to **MIQCP**. Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of A, and let $v_1, \ldots, v_n$ denote the associated eigenvectors. Consider the following reformulation of the above inequality obtained by introducing two auxiliary variables, s_k and y_k, for every negative eigenvalue of A.

$$\begin{aligned}
&\sum_{\lambda_k > 0} \lambda_k \left(v_k^T x\right)^2 + a^T x + b + \sum_{\lambda_k < 0} \lambda_k s_k \le 0,\\
&y_k = v_k^T x \ , \quad \forall\ k:\ \lambda_k < 0,\\
&s_k = y_k^2 \ , \quad \forall\ k:\ \lambda_k < 0.
\end{aligned}$$

Because all variables that are involved in bilinear terms of **MIQCP** are assumed to have non-infinite lower/upper bounds, the same property carries over to auxiliary variables s_k and y_k. By introducing such auxiliary variables for every non-convex constraint $x^T A_k x + a_k^T x + b_k \le 0$ $(k \in M)$, we get the following alternative reformulation of **MIQCP**, referred to as the *eigen reformulation* (**ER**) in the sequel.

$$(\mathbf{ER}) \quad \begin{aligned}
&\min\ a_0^T x\\
&\text{s.t.}\\
&x^T A_k x + a_k^T x + b_k \le 0 \ , \quad \forall k \in M \ ;\\
&x_j \in \mathbb{Z} \ , \quad \forall j \in N_1 \ ;\\
&\textstyle\sum_{\lambda_{kj} > 0} \lambda_{kj} \left(v_{kj}^T x\right)^2 + a_k^T x + b_k + \sum_{\lambda_{kj} < 0} \lambda_{kj} s_{kj} \le 0 \ , \quad \forall k \in M \ ;\\
&y_{kj} = v_{kj}^T x \ , \quad \forall\ j:\ \lambda_{kj} < 0,\ k \in M \ ;\\
&s_{kj} = y_{kj}^2 \ , \quad \forall\ j:\ \lambda_{kj} < 0,\ k \in M \ ;\\
&L_{kj} \le y_{kj} \le U_{kj} \ , \quad \forall\ j:\ \lambda_{kj} < 0,\ k \in M \ .
\end{aligned}$$

For $k \in M$, $\lambda_{k1}, \ldots, \lambda_{kn}$ denote the eigenvalues of A_k and $v_{k1}, \ldots, v_{kn}$ denote the associated eigenvectors. L_{kj} (U_{kj}) are valid lower (upper) bounds on $y_{kj} = v_{kj}^T x$, which can be easily determined by minimizing (maximizing) $v_{kj}^T x$ over a suitably chosen convex relaxation of **MIQCP**. For the sake of brevity, we let $\mathcal{I}$ denote the index set of non-convex constraints of the form $s_{kj} = y_{kj}^2$ in **ER**. Thus for $k \in \mathcal{I}$, **ER** contains auxiliary variables y_k and s_k, and the non-convex constraint $s_k = y_k^2$. The theorem that follows uses polarity to derive strong valid cutting planes for **ER**. This theorem is motivated by a recent application of the same idea in the context of probabilistic programming [18]. For $S \subseteq \mathcal{I}$, we denote by $(< y_k >_{k \in S})$ the sub-vector of y having components indexed by S.

Theorem 5 *Let $S \subseteq \mathcal{I}$ denote a non-empty subset of $\mathcal{I}$, $\mathcal{P}$ denote a polyhedral relaxation of* **ER**, *and let $Q = \{(< y_k >_{k \in S}) \mid \exists x, s, (< y_k >_{k \notin S})$ such that $(x, y, s) \in \mathcal{P}\}$ denote the projection of $\mathcal{P}$ to the space of $(< y_k >_{k \in S})$ variables. Let $V = \{(< y_k^t >_{k \in S}) \mid t = 1 \ldots K\}$ denote the set of extreme points* *of Q. The point $(\hat{x}, \hat{y}, \hat{s}) \in \mathcal{P}$ is a feasible solution to* **ER** *only if the optimal value of the following linear program is non-negative.*

$$
(\textbf{PolarLP}) \quad
\begin{array}{l}
\min \ \sum_{k \in S} (\alpha_k \hat{y}_k + \beta_k \hat{s}_k) - \gamma \\
\text{s.t.} \\
\sum_{k \in S} \left(\alpha_k y_k^t + \beta_k \left(y_k^t\right)^2 \right) - \gamma \geq 0, \quad t = 1 \dots K \ ; \\
\beta_k \leq 0, \quad \forall k \in S \ ; \\
\alpha_k - \alpha_k^+ + \alpha_k^- = 0, \quad \forall k \in S \ ; \\
\sum_{k \in S} \left(\alpha_k^+ + \alpha_k^- - \beta_k \right) = 1 \ ; \\
\alpha_k^+ \geq 0, \ \alpha_k^- \geq 0 \ .
\end{array}
$$

Furthermore, if $(\alpha, \beta, \gamma, \alpha^+, \alpha^-)$ is a feasible solution to **PolarLP** *having negative objective value, then $\sum_{k \in S} (\alpha_k y_k + \beta_k s_k) - \gamma \geq 0$ is valid for* **ER** *and cuts off $(\hat{x}, \hat{y}, \hat{s})$.*

Proof Suppose (x, y, s) is a feasible solution to **ER** and $(\alpha, \beta, \gamma, \alpha^+, \alpha^-)$ is a feasible solution to **PolarLP**. Because $(x, y, s) \in \mathcal{P}$, we have $(< y_k >_{k \in S}) \in Q$ and there exist $\lambda_t \geq 0$ $(t = 1 \dots K)$ such that $\sum_{t=1}^K \lambda_t = 1$ and $y_k = \sum_{t=1}^K \lambda_t y_k^t$ $(k \in S)$. Consequently,

$$
\begin{aligned}
\sum_{k \in S} (\alpha_k y_k + \beta_k s_k) - \gamma &= \sum_{k \in S} \left(\alpha_k \left(\sum_{t=1}^K \lambda_t y_k^t \right) + \beta_k \left(\sum_{t=1}^K \lambda_t y_k^t \right)^2 \right) - \gamma \\
&\geq \sum_{t=1}^N \lambda_t \left(\sum_{k \in S} \left(\alpha_k y_k^t + \beta_k \left(y_k^t \right)^2 \right) - \gamma \right) \quad (\text{because } \beta_k \leq 0, \ \forall k \in S) \\
&\geq 0 \ .
\end{aligned}
$$

□

The derivation of the inequality $\sum_{k \in S} (\alpha_k y_k + \beta_k s_k) - \gamma \geq 0$, referred to as *polarity cut* in the sequel, is based on a three step procedure. The first step is a "projection" step which projects the polyhedral relaxation $\mathcal{P}$ to derive Q. The second step is a "lifting" step which lifts Q to derive

$$
Q_2 = \text{clconv}\left(Q_1 \cup \{ (< y_k >_{k \in S}, < s_k >_{k \in S}) \mid s_k \leq 0 \ \forall k \in S \} \right),
$$

where

$$
Q_1 = \text{clconv}\left(\cup_{t=1}^K \ \{ (< y_k^t >_{k \in S}, < s_k >_{k \in S}) \mid s_k = (y_k^t)^2 \ \forall k \in S \} \right).
$$

The third and final step constructs the polar of Q_2, truncates it with a normalization constraint $\sum_{k \in S} \left(\alpha_k^+ + \alpha_k^- - \beta_k \right) = 1$ and derives the cut generating linear program **PolarLP**. Of these three steps, the second "lifting" step is the most important one for two reasons.

First, it is the only step that performs a non-convex operation. To see this, recall that projection is a linear (and hence convex) operation whereas the polar of a closed convex set Q_2 cannot capture any characteristic that is not already present in Q_2. Consequently, neither the first step nor the last step performs a non-convex operation. The second step, on the other hand, uses a convex function $(f(y_k) = (y_k)^2)$ to lift the set Q to derive Q_1, and then constructs the *hypograph* of the resulting set to derive Q_2. Because the hypograph of a convex function is a non-convex set, it is precisely this step that captures a portion of the non-convexity of **ER**, and lends utility to the above theorem.

Second, generating a valid lifting of Q in the space of $(< y_k >_{k \in S})$ is a non-trivial task. To see this, note that feasible solutions of **ER** need not necessarily project to the extreme points of the set Q. Consequently, it is not guaranteed that every feasible solution to **ER** is contained in the set $Q_1 = \text{clconv}\left(\cup_{t=1}^K \ \left\{ (< y_k^t >_{k \in S}, < s_k >_{k \in S}) \mid s_k = \left(y_k^t\right)^2 \ \forall k \in S \right\} \right)$ obtained by applying the lifting operation to the extreme points of Q. We need an additional device to ensure such a valid lifting, and as Theorem 5 demonstrates, amending Q_1 with the recession cone $\{ (< y_k >_{k \in S}, < s_k >_{k \in S}) \mid s_k \leq 0 \ \forall k \in S \}$ accomplishes exactly that.

As an illustration, consider the special case when S is a singleton, say $S = \{k\}$, and the associated non-convex constraint is $s_k = (y_k)^2$. In this case the projection step is equivalent to determining

PolarLP Solution	Polarity Cut
$\alpha_k = 1,\ \beta_k = 0,\ \gamma = L_k$	$L_k \le -y_k$
$\alpha_k = -1,\ \beta_k = 0,\ \gamma = -U_k$	$-U_k \le -y_k$
$\alpha_k = -\frac{L_k+U_k}{1+L_k+U_k},\ \beta_k = \frac{-1}{1+L_k+U_k},\ \gamma = \frac{L_kU_k}{1+L_k+U_k}$	$\frac{L_kU_k}{1+L_k+U_k} \le \left(\frac{L_k+U_k}{1+L_k+U_k}\right) y_k - \frac{s_k}{1+L_k+U_k}$

Table 1 Illustration of Theorem 5

lower/upper bounds on the y_k variable. The projected set Q is given by $Q = \{y_k \mid L_k \le y_k \le U_k\}$, the set of extreme points is given by $V = \{(L_k), (U_k)\}$ and the polar program is given by,

$$\begin{array}{l} \min\ \alpha_k \hat{y}_k + \beta_k \hat{s}_k - \gamma \\ \text{s.t.} \\ \alpha_k L_k + \beta_k (L_k)^2 - \gamma \ge 0\ ; \\ \alpha_k U_k + \beta_k (U_k)^2 - \gamma \ge 0\ ; \\ \alpha_k - \alpha_k^+ + \alpha_k^- = 0\ ; \\ \alpha_k^+ + \alpha_k^- - \beta_k = 1\ ; \\ \alpha_k^+ \ge 0,\ \alpha_k^- \ge 0,\ \beta_k \le 0\ . \end{array}$$

The above linear program has only three non-trivial basic feasible solutions given in Table 1 along with each of the corresponding polarity cuts. Note that the first two polarity cuts are just bound constraints, whereas the third cut is the secant approximation of the univariate non-convex constraint $s_k \le (y_k)^2$ on the interval $[L_k, U_k]$. Consequently, Theorem 5 can be viewed as generalizing the well-known apparatus of secant approximation based convexification techniques to higher dimensions.

One may be tempted to believe that **PolarLP** can be solved by a row-generation algorithm that works with a subset of extreme points of Q and dynamically generates additional extreme points as needed. Such an approach is unlikely to succeed because the associated separation problem is non-convex and most likely an NP-hard problem itself (see [12]).

Theorem 5 can be generalized to the case where $\mathcal{P}$ is a convex (not necessarily polyhedral) relaxation of **ER** provided that Q is chosen to be a polyhedral outer approximation of the projection of $\mathcal{P}$ to the space of $(< y_k >_{k\in S})$ variables. Such an outer approximation can be generated, for instance, by optimizing various linear functions of the form $\sum_{k\in S} \theta_k y_k$ over the convex relaxation $\mathcal{P}$ of **MIQCP**. In our implementation we chose $\mathcal{P}$ to be the outer approximation of **ER** defined by the incumbent solution, and used all subsets of $\mathcal{I}$ of cardinality two to generate polarity cuts. For each one of these subsets, we computed all of the facets of the projection of $\mathcal{P}$ by solving a family of parametric linear programs over $\mathcal{P}$ using a standard homotopy procedure [14]. These facets were then relaxed by a small amount to derive a numerically stable and "safe" outer approximation of the projected set, and the extreme points of the resulting set were used to construct **PolarLP**.

5 Computational Results

In this section we present our computational results. Because the aim of these experiments was to assess the relative strengths of various relaxations introduced in the previous section, we report the duality gap closed by each one of them along with the time taken to generate the respective relaxation.

Note that all of the results presented in the previous sections pertain to cutting plane generation. In other words, given an incumbent solution $\hat{x}$ to a convex relaxation of **MIQCP**, these sections discuss techniques for generating valid linear and convex quadratic cuts that cut off $\hat{x}$. However, in order to access these cut generators, we need an initial convex relaxation of **MIQCP**; we next address the issue of generating such an initial relaxation. All of our experiments were conducted on the eigen reformulation of **MIQCP**. We used the following convexification of **ER** as our initial convex relaxation of **MIQCP**.

$$
(\textbf{MIQCP-Initial}) \qquad
\begin{array}{l}
\min\ a_0^T x \\
\text{s.t.} \\
\sum_{\lambda_{kj}>0} \lambda_{kj} \left(v_{kj}^T x\right)^2 + a_k^T x + b_k \\
\qquad - \sum_{C_{ij}^k>0} C_{ij}^k y_{ij}^+(x) - \sum_{C_{ij}^k<0} C_{ij}^k y_{ij}^-(x) \le 0, \quad \forall k \in M\ ; \\
\sum_{\lambda_{kj}>0} \lambda_{kj} \left(v_{kj}^T x\right)^2 + a_k^T x + b_k + \sum_{\lambda_{kj}<0} \lambda_{kj} s_{kj} \le 0, \quad \forall k \in M\ ; \\
y_{kj} = v_{kj}^T x\ , \quad \forall\ j:\ \lambda_{kj} < 0,\ k \in M\ ; \\
s_{kj} \ge y_{kj}^2\ , \quad \forall\ j:\ \lambda_{kj} < 0,\ k \in M\ ; \\
s_{kj} - (L_{kj} + U_{kj})\, y_{kj} + L_{kj} U_{kj} \le 0, \quad \forall\ j:\ \lambda_{kj} < 0,\ k \in M\ ; \\
L_{kj}\ \le y_{kj} \le U_{kj}\ , \quad \forall\ j:\ \lambda_{kj} < 0,\ k \in M\ ,
\end{array}
$$

where $C^k = \sum_{\lambda_{kj}<0} (-\lambda_{kj}) \left(v_{kj}^T x\right)^2$ for $k \in M$.

In addition to the cut generators described in the previous sections, we also used the Cut Generating Linear Programming (CGLP) framework described in our companion paper [17] (also see [2]) to derive disjunctive cuts. Recall that the CGLP framework requires a polyhedral relaxation of **MIQCP** and a class of disjunctions that is satisfied by every feasible solution to the problem. Similar to [17], we used the outer approximation of **MIQCP** defined by the incumbent solution to derive a polyhedral relaxation of **MIQCP**. As for the choice of disjunctions, we used the following spatial disjunctions associated with variables x_k appearing in bilinear terms,

$$
\left\{ x \in \mathbb{R}^n\ :\ x_k \le \frac{l_k + u_k}{2} \right\} \bigvee \left\{ x \in \mathbb{R}^n\ :\ x_k \ge \frac{l_k + u_k}{2} \right\}\ .
$$

Furthermore, we strengthened the above disjunction by deriving convex quadratic cuts for each term of the disjunction using the constraints,

$$
\sum_{\lambda_{kj}>0} \lambda_{kj} \left(v_{kj}^T x\right)^2 + a_k^T x + b_k - \sum_{C_{ij}^k>0} C_{ij}^k y_{ij}^+(x) - \sum_{C_{ij}^k<0} C_{ij}^k y_{ij}^-(x) \le 0, \quad \forall k \in M,
$$

and the modified bound on the x_k variable as dictated by the respective term of the disjunction.

We implemented our cut generators using the open source framework Bonmin [6] from COIN-OR. The convex quadratic relaxations were solved using Ipopt [24], eigenvalue problems were solved using Lapack, and all of the linear programs were solved using CPLEX 10.1 . We define the duality gap closed by a relaxation $\mathcal{R}$ of **MIQCP** as $\frac{opt(\mathcal{R})-RLT}{opt-RLT} \times 100$ where $opt(\mathcal{R})$, RLT and opt are the optimal values of $\mathcal{R}$, **MIQCP-RLT** and **MIQCP**, respectively. Note that **MIQCP-RLT** refers to the RLT relaxation of **MIQCP** obtained *without* using the eigen reformulation technique.

Next we describe our computational results on the following three test-beds: GLOBALLib [9], instances from Lee and Grossmann [11], and Box-QP instances from [22].

GLOBALLib is a repository of 413 global optimization instances of widely varying types and sizes. Of these 413 instances, we selected all problems with at most 50 variables that can be easily converted into instances of **MIQCP**. For instance, some of the problems have product-of-powers terms ($x_1x_2x_3x_4x_5$, x_1^3, $x^{0.75}$, etc.) which can be converted into quadratic expressions by introducing additional variables. Additionally, some of the problems do not have explicit upper bounds on the variables; for such problems we used linear programming to determine valid upper bounds thereby making them amenable to techniques discussed in this paper. The final set of selected problems comprised 151 instances.[3]

We implemented the following two variants of our code for the GLOBALLib instances. Both of these variants are cutting planes frameworks that differ in the specific kinds of cutting planes that are used. The first variant uses the disjunctive cut generator described above and the **ProjLP** framework (Section 2) to derive valid inequalities for **MIQCP**. The second variant is identical to the first one, except that it also uses the **PolarLP** framework (Section 4) to derive polarity cuts.

[3] These instances are available in AMPL .mod format from www.andrew.cmu.edu/user/anureets/MIQCP

	W1	**W2**	**V1**	**V2**
>99.99 % gap closed	19	23	16	23
98-99.99 % gap closed	22	31	1	44
75-98 % gap closed	35	33	10	23
25-75 % gap closed	34	23	11	22
0-25 % gap closed	14	14	87	13
0-(-0.22) % gap closed	4	4	0	0
Total Number of Instances	128	128	126	126
Average Gap Closed	70.65%	76.06%	25.59%	79.34%
Average Time taken (sec)	4.616	19.462	198.043	978.140

Table 2 Summary Results: GLOBALLib instances with non-zero Duality Gap

Tables 10–13 describe the computational results. Among the 151 GLOBALLib instances in our test-bed, 23 instances have zero duality gap. Tables 10–12 report the computational results on the remaining 128 instances while Table 2 reports the same in a summarized form. The second column of Tables 10–12 reports the optimal value of **MIQCP-RLT** while the third column reports the value of the best known solution. The next two columns report the duality gap closed by variants 1 and 2 of our code. In our companion paper [17], we proposed various techniques for strengthening the relaxation of **MIQCP** in the extended space obtained by introducing the $Y_{ij} = x_i x_j$ variables. For comparison, we report the duality gaps closed by variants 1 and 2 of the algorithm presented in [17] in the next two columns of the tables, titled V1 and V2, respectively. Variant V1 solves the SDP relaxation **MIQCP-SDP** of **MIQCP** in a cutting plane fashion by using convex quadratic cuts. Variant V2 is identical to V1 except that it also uses disjunctive cuts derived from the non-convex expression $xx^T - Y \not\succeq 0$. The next four columns report the computing times for each one of these four variants. Several comments are in order.

First, both variants 1 and 2 of our code close substantially more gap than variant V1 which corresponds to the SDP relaxation of **MIQCP**. Second, while V2 closes more gap than W1 and W2 it is also computationally more expensive; the average computing times of V2, W1 and W2 are 978.14 sec, 4.616 sec and 19.462 sec, respectively. Third, the strengthened relaxations constructed by V2 are defined in the space of (x, Y) variables and are encumbered with a large number of Y_{ij} variables. Consequently, a branch-and-bound algorithm that uses these relaxations has to bear the computational overhead arising from additional Y_{ij} variables at every node of the branch-and-bound tree. The strengthened relaxations constructed by variants W1 and W2, on the other hand, are defined only in the space of x variables and are hence much more desirable for a branch-and-bound algorithm. Fourth, variant W2 closes at least 10% more duality gap than variant W1 on 19 instances (ex2_1_5, ex3_1_2, himmel11, st_glmp_kky, st_kr, st_ph15, etc.) thereby demonstrating the marginal importance of polarity cuts.

In order to assess the performance of our code on 23 instances with no duality gap, we report the maximum infeasibility $\max_{k \in \mathcal{I}} \left(\hat{s}_k - \hat{y}_k^2\right)$ in Table 10 for these instances, where $(\hat{x}, \hat{y}, \hat{s})$ denotes the solution of the convex relaxation at the last iteration of the respective variant. It is interesting to note that both variants of our code were able to produce almost feasible solutions to 14 out of 23 instances.

The ex9$\star$ instances in the GLOBALLib repository contain the linear-complementarity constraints (LCC) $x_i x_j = 0$ on a subset of variables. These constraints give rise to the following disjunction, $(x_i = 0) \vee (x_j = 0)$, which in turn can be embedded within the CGLP framework to generate disjunctive cuts. In order to test the effectiveness of these cuts, we modified our code to automatically detect linear-complementarity constraints and used the corresponding disjunctions along with the default medley of disjunctions to generate disjunctive cuts. Table 13 reports our computational results. We observe that while the default version of our code is unable to close any significant gap on the ex9_1_4 instance, when augmented with disjunctive cuts from the linear-complementarity constraints, it closes 100% of the duality gap.

Note that among the three cut generators used by variants W1 and W2, namely **ProjLP**, **PolarLP** (only W2) and CGLP, the disjunctive cut generator CGLP is computationally most expensive because it requires solving large highly degenerate linear programs. In order to evaluate the marginal contribution of disjunctive cuts, we conducted the following experiment on 128 GLOBALLib instances with non-

	W1	W2	W1-Dsj	W2-Dsj
>99.99 % gap closed	19	23	19	23
98-99.99 % gap closed	22	31	5	21
75-98 % gap closed	35	33	17	18
25-75 % gap closed	34	23	26	32
0-25 % gap closed	14	14	57	30
0- (-0.22) % gap closed	4	4	4	4
Total Number of Instances	128	128	128	128
Average Gap Closed	70.65%	76.06%	40.92%	60.48%
Average Time taken (sec)	4.616	19.462	0.893	0.814

Table 3 Marginal Value of Disjunctive Cuts

zero duality gap. We modified our code so that the disjunctive cut generator was turned off for each one of the variants W1 and W2. Tables 15-17 report the resulting computational results while Table 3 reports the same in summarized form. A suffix of "-Dsj" indicates that the corresponding version of our code was modified to not use the disjunctive cut generator. Three remarks are in order.

First, switching off the disjunctive cut generator adversely affects the performance of both the variants, as expected. However, the degradation in average duality gap closed is much higher for W1 (around 30%) than for W2 (around 16%), suggesting that polarity cuts are able to capture a certain portion of the strengthening that is derived from disjunctive cuts. Second, the average computing times for variants W1 and W2 without disjunctive cuts are less than 1 sec, thus demonstrating their practical utility as computationally efficient strengthening techniques. Third, it is interesting to examine the source of strengthening for the W1-Dsj variant. Note that the only cut generator used by W1-Dsj is **ProjLP** which in turn is a device to project the **MIQCP-RLT** formulation to the space of x-variables. In the absence of any other cut generator, what is aiding W1-Dsj to the extent that it closes 40% of the duality gap on average? The answer to this question lies in our use of eigen reformulation. Recall that eigen reformulation entails introducing additional variables $y_j,\ s_j\ (j \in \mathcal{I})$ which are derived from eigenvectors of A_k $(k \in M)$ matrices with negative eigenvalues, and keeping the convex quadratic terms corresponding to positive eigenvalues. Alternatively, our initial formulation **MIQCP-Initial** identifies directions of maximal non-convexity in each constraint, introduces additional variables to expose them and then relaxes the non-convex constraint $s_k \leq (y_k)^2$ to its secant approximation $s_k \leq (L_k + U_k)y_k - L_kU_k$ to create a convex relaxation. For the convex side of the constraints, **MIQCP-Initial** identifies the directions of convexity of each constraint and preserves them thereby capturing a portion of strengthening derivable from **MIQCP-SDP**. It is precisely this specific way of lifting the **MIQCP** formulation that explains the 40% duality gap closed by W1-Dsj variant of our code.

Next we present our computational results on the **MIQCP** instances proposed in [11]. These problems have both continuous and integer variables and quadratic constraints. They are of relatively small size with between 10 and 54 variables. All of these problems contain the so-called SOS1 constraint of the form $x_1 + x_2 + x_3 = 1$, where $x_1,\ x_2,\ x_3$ are binary variables. These SOS1 constraints imply the following disjunction, $(x_1 = 1) \vee (x_2 = 1) \vee (x_3 = 1)$ which in turn can be used within the CGLP framework to generate disjunctive cuts. We modified our code to automatically detect such SOS1 constraints, and use the corresponding disjunctions along with the default medley of disjunctions to generate disjunctive cuts. Table 4 summarizes the experiment. Note that both variants of our code out-perform V1. However, unlike the case of GlobalLib instances, the V2 variant of the algorithm presented in [17] perform significantly better on these instances than W1 or W2.

Next we present our results on the box-constrained Quadratic Programs (QPs). This test bed consists of test problems used in [22]. These problems are randomly generated box QPs with A_0 of various densities. Similar to GLOBALLib instances, we ran both variants of our code on all of these instances, and we also performed additional experiments to determine the marginal impact of disjunctive and polarity cuts. Based on our computational results, we conclude that disjunctive and polarity cuts have inconsequential effect on the fraction of the duality gap closed for these instances. Alternatively, all four variants of our code, W1, W2, W1-Dsj and W2-Dsj, close more or less the same duality gap.

			Relaxation Values			
Instance	**RLT**	**OPT**	**W1**	**W2**	**V1**	**V2**
Example 1	-58.70	-11.00	-37.44	-37.44	-58.70	-37.44
Example 2	-414.94	-14.00	-57.74	-57.74	-93.18	-14.26
Example 3	-819.66	-510.08	-603.38	-601.32	-793.15	-513.61
Example 4	-499282.59	-116575.00	-467386.78	-461639.29	-472727.49	-363487.69

Table 4 Summary of results on the Lee-Grossmann examples.

Note that box constrained QPs can be cast as MIQCPs with a single non-convex quadratic constraint thereby making them amenable to the projected subgradient heuristic discussed in Section 3. Based on this observation, we designed a third variant W3 of our code which uses the **ProjLP** framework and the projected subgradient heuristic to generate linear and convex quadratic cuts. We ran this variant on all the box QP instances described in [22]; Tables 18 and 19 report the computational results while Table 5 reports the same in summarized form. In order to evaluate the marginal contribution of convex quadratic cuts derived via the projected subgradient heuristic, we ran a modified version of W3 wherein the cut generator for these cuts was switched off; Tables 18, 19 and 5 report the computational results of this experiment in columns titled W3-SDP.

	% Duality Gap Closed		Time Taken (sec)			%Time spent on Cut Generation		Time (sec) to solve last relaxation	
Instance	W3	W3-SDP	W3	W3-SDP	W3 (Adj)	W3	W3-SDP	W3	W3-SDP
spar20*	94.60 - 99.97	91.54 - 99.91	2.49 - 408.36	0.84 - 2.46	0.51 - 1.60	26.28 - 95.77	0.12 - 0.24	0.05 - 0.33	0.01 - 0.09
spar30*	89.87 - 99.99	51.41 - 98.79	12.33 - 565.88	1.74 - 14.38	3.32 - 14.49	17.78 - 91.48	0.00 - 0.21	0.07 - 0.9	0.01 - 0.23
spar40*	87.85 - 99.60	21.78 - 89.63	35.77 - 134.8	4.16 - 65.28	13.75 - 49.76	27.5 - 78.37	0.01 - 0.13	0.16 - 1.19	0.02 - 0.75
spar50*	87.88 - 97.53	11.38 - 50.15	50.22 - 180.96	8.76 - 99.13	28.95 - 76.19	51.02 - 79.73	0.01 - 0.11	0.13 - 0.87	0.04 - 1.01
spar60*	85.78 - 90.99	0.00 - 0.00	121.83 - 226.11	111.07 - 127.47	86.28 - 141.77	46.29 - 56.61	0.10 - 0.12	0.54 - 1.55	1.65 - 2.17
spar70*	89.78 - 99.36	0.00 - 53.67	191.12 - 693.28	22.02 - 202.98	92.63 - 143.35	71.13 - 87.7	0.01 - 0.11	0.48 - 1.1	0.08 - 2.42
spar80*	88.13 - 97.49	2.94 - 56.23	257.62 - 892.96	34.77 - 67.66	121.62 - 230.53	76.37 - 84.44	0.01 - 0.02	0.57 - 2.03	0.1 - 0.82
spar90*	89.44 - 96.60	5.73 - 50.13	408.73 - 991.04	46.98 - 95.66	184.63 - 294.92	73.44 - 88.25	0.01 - 0.02	0.78 - 1.51	0.12 - 2.11
spar100*	92.15 - 96.46	8.17 - 51.79	538.03 - 1509.96	75.49 - 112.69	279.41 - 385.64	77.49 - 92.3	0.01 - 0.23	0.82 - 2.01	0.13 - 2.5
Average	**95.19%**	**50.01%**	**280.50**	**37.89**	**101.57**	**66.05%**	**0.04%**	**0.67**	**0.33**

Table 5 Summary Results: Box Constrained QPs from [22]

The second column of Tables 18 and 19 reports the optimal value of the **MIQCP-RLT** relaxation while the third column reports the optimal value of each instance. The next two columns report the duality gap closed by W3 and W3-SDP, respectively, while the following two columns report the total computing time for each variant. Like most cutting plane algorithms, variant W3 exhibits a strong tailing off behavior (i.e., most of the duality gap is closed in the first few iterations whereas the ensuing iterations contribute very little). In order to quantitatively assess the impact of this tailing off phenomenon on the total computing time, we computed the time it takes for the W3 variant to close a fraction of the duality gap that is 1% less than the duality gap closed in its entire run. The eighth column of Tables 18 and 19 titled "W1 (Adj)" reports the resulting computing times. For the sake of illustration consider the spar100-075-1 instance. Variant W3 closes 95.84% of the duality gap on this instance in 1509.36 sec; however it takes only 366.24 sec to close 94.84% of the gap. In other words, the code was able to close a significant proportion of the duality gap in the first 6 min. of the experiment, while the last 19 min. was spent on generating cuts that closed just 1% more duality gap.

The next two columns of the tables report the fraction of the total computing time that was spent on cut generation. Two remarks are in order. First, the fraction of time spent on cut generation in variant W3 increases as the problem size increases. For larger instances with more than 85 variables, almost 75% of the computational effort was spent on cut generation. Second, the same statistic for variant W3-SDP is significantly smaller and never goes beyond 0.25%. This can be attributed to the fact that the only cut generator used by W3-SDP is **ProjLP** which involves solving linear programs (**DProjLP**) with a lot of variables but very few constraints (also see discussion in Section 2). These statistics attest the practical usefulness of the **ProjLP** framework.

The last column of Tables 18 and 19 reports the time taken by Ipopt to solve the final strengthened relaxation for each instance. Note that the strengthened relaxations of even the larger instances with 100 variables can be solved in less than 3 sec. This observation has an interesting consequence which we discuss next. Recall that there are two critical issues that are involved in engineering an efficient branch-and-bound algorithm, namely, the strength of the relaxation used and the computational effort spent on solving it at every node of the branch-and-bound tree. Naturally we are interested in relaxations that are a good representation of the convex hull of all feasible solutions (i.e., have small duality gaps), and which can be solved efficiently. Tables 18 and 19 show that the strengthened relaxations obtained by our algorithm have both of these desirable properties. These relaxations close around 95% (average) of the duality gap and can be solved in less than a second on average. To give a better appreciation of this phenomenon to the reader, we conducted the following experiment. We chose two state-of-the-art SDP solvers, SDPLR [7] and SDPA [25], and solved the SDP relaxation **MIQCP-SDP** of these box QP instances using them. Tables 20 and 21 report the computational results, while Table 6 reports the same in summarized form. Four remarks are in order.

First, the amount of computational effort required to solve the strengthened relaxations (last column of Tables 20 and 21) is several orders of magnitude smaller than the one required to solve the SDP relaxation using a black box SDP solver. This observation naturally accrues significance in view of the fact that such a relaxation has to be solved hundreds or thousands of times in a branch-and-bound procedure. Second, the time spent on *generating* the strengthened relaxation is comparable and in most cases less than the time required to solve the SDP relaxation. Note that most contemporary branch-and-bound procedures generate cutting planes primarily at the root node and only sparingly at other nodes of the branch-and-bound tree. Consequently, the amortized cost of generating the strengthened relaxation decreases as the number of branch-and-bound nodes increases. Third, the convex quadratic cuts generated using the projected subgradient heuristic come close to capturing the strength of the SDP relaxation, the heuristic nature of their separation routine notwithstanding. Of course, for larger instances the gap between the strength of the projected relaxation and the extended SDP relaxation widens, highlighting the heuristic nature of our approach. Fourth, the convex quadratic constraints in the strengthened relaxations generated by our code can be approximated by polyhedral relaxations introduced by Ben-Tal and Nemirovski [5] (also see [23]) yielding linear programming (LP) relaxations of these problems. Such LP relaxations are extremely desirable for branch-and-bound algorithms for two reasons. One, they can be efficiently re-optimized using warm-starting capabilities of LP solvers thereby reducing the computational overheads at nodes of the enumeration tree. Two, these LP relxations can easily avail techniques, such as branching strategies, cutting planes, heuristics, etc., which have been developed by the MILP community in the past five decades (see [1] for application of these techniques in the context of convex MINLPs).

Indeed, one could argue that the SDP solvers can be engineered to efficiently handle the large number of RLT inequalities (for example [15]) thereby improving the rather grim picture presented in Tables 20 and 21. Furthermore, instead of solving the SDP relaxation to optimality, the optimization process can be pre-empted to improve the overall computing times. Despite these engineering improvements, its unlikely that one can obtain relaxations of **MIQCP** in the space of (x, Y) variables that are at least as strong as the relaxations proposed in this paper and can be solved with as little computing effort as documented in the last column of Tables 18 and 19. Table 7 gives detailed statistics on some of the larger instances to demonstrate this computational chasm between the extended SDP relaxations and those proposed in this paper (labelled "Proj" in the table).

	% Duality Gap Closed			Time Taken (sec)			Time to solve last relaxation (sec)
Instance	**SDPLR**	**SDPA**	**W3**	**SDPLR**	**SDPA**	**W3**	**W3**
spar20*	99.67 - 100	99.67 - 99.99	94.6 - 99.97	0.97 - 56.37	1.98 - 3.39	2.48 - 408.35	0.05 - 0.32
spar30*	97.81 - 100	97.81 - 99.99	89.87 - 99.99	3.57 - 243.3	16.66 - 29.33	12.33 - 565.88	0.06 - 0.89
spar40*	96.6 - 100	96.6 - 99.99	87.85 - 99.6	10.3 - 515.73	105.68 - 157.83	35.77 - 134.8	0.16 - 1.18
spar50*	95.55 - 100	95.55 - 99.99	87.88 - 97.53	41.72 - 926.15	438.77 - 589.17	50.21 - 180.95	0.13 - 0.86
spar60*	98.69 - 100	98.69 - 99.99	85.78 - 90.99	88.05 - 532.45	1150.06 - 1408.32	121.83 - 226.1	0.53 - 1.55
spar70*	98.46 - 100	98.46 - 99.99	89.78 - 99.36	133.07 - 3600.75	2769.98 - 3721.34	191.11 - 693.27	0.48 - 1.1
spar80*	97.85 - 100	97.84 - 99.99	88.13 - 97.49	965.18 - 5413.02	6618.79 - 8285.12	257.61 - 892.95	0.56 - 2.02
spar90*	97.83 - 99.99	97.83 - 99.99	89.44 - 96.6	2403.62 - 7049.49	12838.46 - 17048.98	408.73 - 991.04	0.77 - 1.51
spar100*	98.17 - 99.38	98.17 - 99.38	92.15 - 96.46	5355.2 - 10295.88	23509.13 - 28604.12	538.02 - 1509.96	0.82 - 2
Average	**99.40%**	**99.40%**	**95.19%**	**1741.20**	**5247.04**	**280.50**	**0.67**

Table 6 Summary Results: Comparison with SDP Solvers

	No. Constraints										
	No. Variables		Linear		Convex (Non-Linear)		Computing Time (sec)		% Duality Gap Closed	
Instances	SDP	Proj	SDP	Proj	SDP ($Y - xx^T \succcurlyeq 0$)	Proj (Quadratic)	SDP	Proj	SDP	Proj
spar100-025-1	5151	203	20201	156	1	119	5719.42	1.14	98.93%	92.36%
spar100-025-2	5151	201	20201	151	1	95	10185.65	1.52	99.09%	92.16%
spar100-025-3	5151	201	20201	150	1	114	5407.09	1.24	99.33%	93.26%
spar100-050-1	5151	201	20201	150	1	98	10139.57	1.07	98.17%	93.62%
spar100-050-2	5151	201	20201	150	1	113	5355.20	1.26	98.57%	94.13%
spar100-050-3	5151	201	20201	150	1	97	7281.26	0.82	99.39%	95.81%
spar100-075-1	5151	201	20201	150	1	131	9660.79	2.00	99.19%	95.84%
spar100-075-2	5151	201	20201	150	1	109	6576.10	1.23	99.18%	96.47%
spar100-075-3	5151	199	20201	147	1	90	10295.88	0.87	99.19%	96.06%

Table 7 Comparison with SDP Solvers (spar100 Instances)

Eigen Vector	Eigen Values
$\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0\right)$	$\frac{1}{2}$
$\left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, 0\right)$	$-\frac{1}{2}$
$\left(0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$	$\frac{1}{2}$
$\left(0, 0, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)$	$-\frac{1}{2}$

Table 8 Eigenvectors and Eigenvalues of A matrix (Theorem 5 Illustration)

We conclude this section by illustrating Theorem 5 on the st_glmp_kky instance from GLOBALLib shown below.

$$
(\textbf{st\_glmp\_kky}) \qquad
\begin{array}{lrr}
\min z & & \\
\text{s.t.} & & \\
\quad x_5x_4 + x_7x_6 + x_3 + -z & \leq & 0 \\
\quad -5x_1 + 8x_2 & \leq & 24 \\
\quad -5x_1 - 8x_2 & \leq & 100 \\
\quad -6x_1 + 3x_2 & \leq & 100 \\
\quad -4x_1 - 5x_2 & \leq & -10 \\
\quad 5x_1 - 8x_2 & \leq & 100 \\
\quad 5x_1 + 8x_2 & \leq & 44 \\
\quad 6x_1 - 3x_2 & \leq & 15 \\
\quad 4x_1 + 5x_2 & \leq & 100 \\
\quad 3x_1 - 4x_2 - x_3 & = & 0 \\
\quad x_1 + 2x_2 - x_4 & = & 1.5 \\
\quad 2x_1 - x_2 - x_5 & = & -4 \\
\quad x_1 - 2x_2 - x_6 & = & -8.5 \\
\quad 2x_1 + x_2 - x_7 & = & 1 \\
\quad 0 \leq x_1 \leq 10.0 & & \\
\quad 0 \leq x_2 \leq 10.0 & & \\
\quad -12 \leq x_3 \leq 7.5 & & \\
\quad 1 \leq x_4 \leq 9 & & \\
\quad 1 \leq x_5 \leq 9 & & \\
\quad 2 \leq x_6 \leq 11 & & \\
\quad 1 \leq x_7 \leq 10 & & .
\end{array}
$$

st_glmp_kky has exactly one non-convex constraint, $x_4x_5 + x_6x_7 + x_3 - z \leq 0$ with Hessian matrix

$$
A = \begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & 0 \end{bmatrix} .
$$

Note that for the sake of brevity we show only the rows and columns of the Hessian matrix corresponding to the nonlinear variables $x_4,\ x_5,\ x_6$ and x_7. Table 8 gives the eigenvectors and eigenvalues of the A matrix.

Clearly, A has two negative eigenvalues, and we can derive the eigen reformulation of st_glmp_kky by introducing four additional variables, say y_1, y_2, s_1 and s_2, and augmenting the original formulation with the constraints:

$$\begin{aligned}
\tfrac{1}{2}\left(\tfrac{1}{\sqrt{2}}x_4+\tfrac{1}{\sqrt{2}}x_5\right)^2+\tfrac{1}{2}\left(\tfrac{1}{\sqrt{2}}x_6+\tfrac{1}{\sqrt{2}}x_7\right)^2+x_3-z-\tfrac{1}{2}s_1-\tfrac{1}{2}s_2 &\le 0\\
\tfrac{1}{\sqrt{2}}x_4-\tfrac{1}{\sqrt{2}}x_5-y_1 &= 0\\
\tfrac{1}{\sqrt{2}}x_6-\tfrac{1}{\sqrt{2}}x_7-y_2 &= 0\\
s_1 &= y_1^2\\
s_2 &= y_2^2\\
-5.6569 \le y_1 &\le 3.7123\\
-3.7123 \le y_2 &\le 4.9497\ .
\end{aligned}$$

The bounds on y_1 and y_2 variables were determined by maximizing and minimizing y_1 and y_2, respectively, over a suitably chosen outer approximation of st_glmp_kky, referred to as OA in the sequel. We can derive the projection of OA to the space of (y_1, y_2) variables by optimizing parametric functions of the form $\theta_1 y_1 + \theta_2 y_2$ over OA; Table 5 reports the facial characteristics of the projected set, say Q.

Facets
$y_2 + 0.4117y_1 \le 2.6205$
$y_2 + 1.0009y_1 \le 2.8310$
$y_2 + 3.2867y_1 \le 8.4890$
$y_2 + 0.3043y_1 \ge -2.5826$
$y_2 + 1.4009y_1 \ge -2.9757$

Extreme Points
$y_1 = 3.7123$ $y_2 = -3.7125$
$y_1 = -0.3584$ $y_2 = -2.4735$
$y_1 = -5.657$ $y_2 = 4.9498$
$y_1 = 0.3571$ $y_2 = 2.4734$
$y_1 = 2.4754$ $y_2 = 0.3531$

Table 9 Facial Characterization of the Projected Set (Theorem 5 Illustration)

The extreme points of Q can be used to derive polarity cuts as explained in Section 4. Variants of our code that use these polarity cuts, namely W2 and W2-Dsj, close 99.62% of the duality gap on the st_glmp_kky instance. On the other hand, variants W1 and W1-Dsj, which do not use polarity cuts are unable to close any gap. Figure 5 provides an explanation for this disparate behavior. The solid lines in the figure plot the facets of the projected set Q, whereas the dotted lines denote the box determined by the lower/upper bounds on y_1 and y_2.

The key to understanding this disparity lies in recognizing the interdependent nature of y_1 and y_2 variables which arises by virtue of constraints that are present in st_glmp_kky. These constraints restrict the set of values that y_1 and y_2 can take *simultaneously*. For instance, even though both y_1 and y_2 can attain their maximum values of 3.7123 and 4.9497 over different feasible solutions, Figure 5 shows that they can never attain these values simultaneously at any feasible solution. It is precisely this global information that is captured by the projection mechanism, and effectively utilized by the **PolarLP** framework to generate strong convex relaxations via polarity cuts from the pair of non-convex constraints $s_1 \le y_1^2$ and $s_2 \le y_2^2$.

6 Generalization to non-convex MINLPs

Interestingly, many of the ideas presented in this paper can be used to generate strong convex relaxations of non-convex Mixed Integer Non-Linear Programs (**MINLP**). For the sake of illustration consider the following **MINLP**,

$$\begin{aligned}
&\min\ x+y-z\\
&\text{s.t.}\\
&xy\ \cos(x-y)\ +\ ze^{x}\ +\ \log(x) \le 3\\
&y^2\ +\ zx \le 2\\
&1 \le y \le 10\\
&0.2 \le x \le 2\\
&-10 \le z \le 10\\
&z \in \mathbb{Z}\ .
\end{aligned}$$

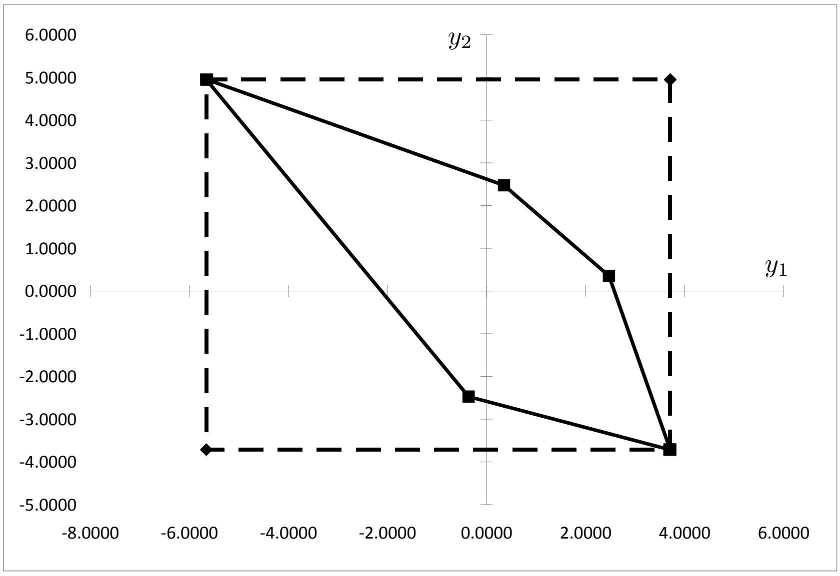

Figure 1 2-Dimensional Projection (Theorem 5 Illustration)

By introducing additional variables we can reformulate the above **MINLP** as,

$$
\begin{aligned}
&\min\ x + y - z \\
&\text{s.t.} \\
&V_{xy}C_{xy} + zE_x + L_x \le 3 \\
&y^2 + zx \le 2 \\
&V_{xy} - xy = 0 \\
&0.2 \le x \le 2,\ 1 \le y \le 10,\ -10 \le z \le 10 \\
&0.2 \le V_{xy} \le 20,\ -1 \le C_{xy} \le 1,\ \mathrm{e}^{0.2} \le E_x \le \mathrm{e}^2 \\
&z \in \mathbb{Z} \\
&C_{xy}\ -\ \cos(x - y) = 0 \\
&L_x\ -\ \log(x) \ge 0 \\
&E_x - \mathrm{e}^x = 0\ .
\end{aligned}
$$

The first six constraints of this reformulation give a **MIQCP** relaxation of the original **MINLP** which is readily amenable to techniques discussed in this paper. For example, we can use the projected subgradient heuristic to approximate the SDP relaxation of the constraint $V_{xy}C_{xy} + zE_x + L_x \le 3$. Similarly, given a convex relaxation, say $\mathcal{R}$, of the above reformulation we can determine a polyhedral outer approximation, say OA, of the projection of $\mathcal{R}$ to the space of (L_x, E_x) variables by optimizing parametric linear functions of the form $\theta_1 L_x + \theta_2 E_x$ over $\mathcal{R}$. The extreme points of OA can be used to derive polarity cuts using a straightforward generalization of Theorem 5.

To summarize, even though the results presented in this paper focussed on **MIQCP**s, they are equally applicable to a much wider class of non-convex **MINLP**s. All we need is an automatic system that can take a non-convex **MINLP** and extract a corresponding **MIQCP** relaxation. Development of software such as Couenne [4,8] is a step in this direction.

Acknowldegments

Research of the first author was supported by the National Science Foundation through grant DMI-0352885 and by the Office of Naval Research through contract N00014-03-1-0133, while he was a graduate student at Tepper School of Business, Carnegie Mellon University. We are grateful to Sam Burer for providing us the box-QP instances and their **MIQCP-SDP** formulations.

References

1. K. Abhishek, S. Leyffer, J. T. Linderoth, Filmint: An outer-approximation-based solver for nonlinear mixed integer programs. Preprint ANL/MCS-P1374-0906, Argonne National Laboratory, Mathematics and Computer Science Division, Argonne, IL, 2006.
2. E. Balas, Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics* 89(1-3), 1998, 3–44.
3. E. Balas. Projection and lifting in combinatorial optimization. In: M. Juenger and D. Naddef (editors), “Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions,” Springer, *Lecture Notes in Computer Science,* volume 2241, 2001, 26–56.
4. Pietro Belotti, Jon Lee, Leo Liberti, Francois Margot, A. Wächter. Branching and bounds tightening techniques for non-convex MINLP, *IBM Research Report* RC24620, 08/2008.
5. A. Ben-Tal and A. Nemirovski. On polyhedral approximations of the second-order cone. *Math. Oper. Res.* 26, 2001, 193–205.
6. 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, 2008, 186–204.
7. S. Burer and R.D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming* (series B) 95, 2003, 329–357.
8. Couenne, `projects.coin-or.org/Couenne`.
9. GLOBALLib, `www.gamsworld.org/global/globallib/globalstat.htm`
10. S. Kim and M. Kojima. Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optimization Methods and Software* 15, 2001, 201–204.
11. S. Lee and I.E. Grossmann. A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems. *Computers and Chemical Engineering* 25, 2001, 1675–1697.
12. T. Matsui. NP-hardness of linear multiplicative programming and related problems. *Journal of Global Optimization* 9, 1996, 113–119.
13. G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I Convex underestimating problems. *Mathematical Programming* 10, 1976, 147–175.
14. J.L. Nazareth, The homotopy principle and algorithms for linear programming, *SIAM Journal on Optimization* 1, 1991, 316–332.
15. F. Rendl, G. Rinaldi and A. Wiegele. A branch and bound algorithm for Max-Cut based on combining semidefinite and polyhedral relaxations. In ”Integer programming and combinatorial optimization (Cornell 2007)”, M. Fischetti, D. P. Williamson Eds., *L*ecture Notes in Computer Science, volume 4513, pages 295–309. Springer-Verlag, Berlin-Heidelberg, 2007.
16. A. Saxena, P. Bonami and J. Lee. Disjunctive Cuts for Non-Convex Mixed Integer Quadratically Constrained Problems. In: ”Integer programming and combinatorial optimization (Bertinoro, 2008),” A. Lodi, A. Panconesi, and G. Rinaldi, Eds., *Lecture Notes in Computer Science,* volume 5035, pp. 17-33. Springer-Verlag Berlin Heidelberg, 2008.
17. A. Saxena, P. Bonami and J. Lee. Convex relaxations of mixed integer quadratically constrained programs: Extended formulations, *IBM Research Report* RC24621, 08/2008.
18. A. Saxena, V. Goyal and M. Lejeune. MIP Reformulations of the probabilistic set covering problem, To appear in *Mathematical Programming.*
19. H.D. Sherali and W.P. Adams. A reformulation-linearization technique for solving discrete and continuous nonconvex problems. Kluwer, Dordecht 1998.
20. R. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming, *Mathematical Programming* 86, 1999, 515–532.
21. D. Vandenbussche and G.L. Nemhauser. A polyhedral study of nonconvex quadratic programs with box constraints. *Mathematical Programming* 102(3), 2005, 531–556.
22. D. Vandenbussche and G.L. Nemhauser. A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. *Mathematical Programming* 102(3), 2005, 559–575.
23. J.P. Vielma, S. Ahmed and G.L. Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed integer conic quadratic programs, *INFORMS Journal on Computing* 20, 2008, 438–450.
24. 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), 2006, 25–57.
25. M. Yamashita, K. Fujisawa, and M. Kojima. Implementation and evaluation of SDPA 6.0 (SemiDefinite Programming Algorithm 6.0). *Optimization Methods and Software* 18, 2003, 491-505.

Appendix

			% Duality Gap Closed				Time Taken (sec)			
Instance	**RLT**	**OPT**	**W1**	**W2**	**V1**	**V2**	**W1**	**W2**	**V1**	**V2**
alkyl	-2.76	-1.77	81.87%	83.42%	0.00%	55.83%	3.766	5.708	10.621	3619.874
circle	0.00	4.57	91.48%	91.48%	45.74%	99.89%	0.228	0.254	0.218	0.456
dispatch	3101.28	3155.29	100.00%	100.00%	100.00%	100.00%	0.021	0.052	0.044	0.052
ex2_1_1	-18.90	-17.00	99.96%	99.96%	0.00%	72.62%	1.265	1.249	0.009	704.400
ex2_1_10	39668.06	49318.02	89.23%	89.23%	22.05%	99.37%	0.071	0.142	6.719	29.980
ex2_1_5	-269.45	-268.01	77.09%	99.95%	0.00%	99.98%	0.049	0.097	0.020	0.173
ex2_1_6	-44.40	-39.00	99.97%	99.96%	0.00%	99.95%	0.385	0.265	0.023	3397.650
ex2_1_7	-6031.90	-4150.41	69.74%	88.09%	0.00%	41.17%	11.179	221.363	0.188	3607.439
ex2_1_8	-82460.00	15639.00	99.64%	99.90%	0.00%	84.70%	0.645	1.692	0.491	3632.275
ex2_1_9	-2.20	-0.38	93.33%	93.96%	0.00%	98.79%	1.094	0.985	0.140	1587.940
ex3_1_1	2533.20	7049.25	0.35%	0.35%	0.00%	15.94%	0.186	0.176	1.391	3600.268
ex3_1_2	-30802.76	-30665.54	36.44%	65.03%	49.74%	99.99%	1.719	1.853	0.035	0.083
ex3_1_3	-440.00	-310.00	99.23%	100.00%	0.00%	99.99%	0.042	0.084	0.013	0.064
ex3_1_4	-6.00	-4.00	26.98%	30.22%	0.00%	86.31%	0.403	0.121	0.009	21.261
ex4_1_1	-173688.80	-7.49	99.71%	99.88%	100.00%	100.00%	0.577	0.452	0.287	0.310
ex4_1_3	-7999.46	-443.67	84.32%	85.89%	56.40%	93.54%	1.318	0.968	0.080	0.285
ex4_1_4	-200.00	0.00	50.00%	50.00%	100.00%	100.00%	0.326	0.425	0.247	0.243
ex4_1_6	-24075.00	7.00	62.08%	62.08%	100.00%	100.00%	0.187	0.177	0.185	0.308
ex4_1_7	-206.25	-7.50	94.86%	94.72%	100.00%	100.00%	1.391	0.832	0.128	0.114
ex4_1_8	-29.00	-16.74	100.00%	100.00%	100.00%	100.00%	0.042	0.036	0.043	0.059
ex4_1_9	-6.99	-5.51	0.15%	6.27%	0.00%	43.59%	0.085	0.121	0.008	1.307
ex5_2_2_case1	-599.90	-400.00	-0.22%	-0.22%	0.00%	0.00%	0.150	0.209	0.011	0.016
ex5_2_2_case2	-1200.00	-600.00	-0.05%	-0.05%	0.00%	0.00%	0.153	0.179	0.021	0.047
ex5_2_2_case3	-875.00	-750.00	-0.02%	-0.01%	0.00%	0.36%	0.071	0.155	0.016	0.358
ex5_2_4	-2933.33	-450.00	67.42%	73.30%	0.00%	79.31%	1.345	1.448	0.046	68.927
ex5_3_2	1.00	1.86	0.00%	0.00%	0.00%	7.27%	1.754	2.665	0.355	245.821
ex5_4_2	2598.25	7512.23	0.52%	0.52%	0.00%	27.57%	0.209	0.177	1.141	3614.376
ex7_3_2	0.00	1.09	52.16%	52.16%	0.00%	59.51%	0.478	0.569	0.788	3609.704
ex8_1_4	-13.00	0.00	51.54%	51.54%	100.00%	100.00%	0.186	0.175	0.020	0.038
ex8_1_5	-3.33	0.00	25.72%	24.41%	68.30%	68.97%	23.192	8.681	0.839	1.246
ex8_1_7	-757.58	0.03	86.11%	87.69%	77.43%	77.43%	11.749	5.521	75.203	75.203
ex8_1_8	-0.85	-0.39	66.29%	65.42%	0.00%	76.49%	7.411	9.191	7.722	3607.682
ex8_4_1	-5.00	0.62	89.05%	89.24%	91.84%	91.09%	3.050	4.913	3659.232	3642.131
ex8_4_2	-5.00	0.49	91.21%	91.41%	94.07%	93.04%	3.100	6.547	3641.875	3606.071
ex9_1_4	-63.00	-37.00	0.00%	0.00%	0.00%	0.00%	0.057	0.256	0.077	0.603
ex9_2_1	-16.00	17.00	68.14%	68.74%	54.54%	60.04%	1.107	0.563	3603.428	2372.638
ex9_2_2	-50.00	100.00	89.62%	93.95%	70.37%	88.29%	0.180	0.276	1227.898	3606.357
ex9_2_3	-30.00	0.00	0.00%	0.00%	0.00%	0.00%	0.237	0.451	0.125	3.819
ex9_2_4	-396.00	0.50	99.87%	99.87%	99.87%	99.87%	0.062	0.076	2.801	8.897
ex9_2_6	-406.00	-1.00	99.88%	99.88%	87.23%	87.93%	0.279	0.436	851.127	2619.018
ex9_2_7	-9.00	17.00	59.30%	61.34%	42.31%	51.47%	0.468	0.899	3602.364	3628.249
ex9_2_8	0.50	1.50	100.00%	100.00%	-	-	0.019	0.037	-	-
himmel11	-30802.76	-30665.54	26.51%	59.00%	49.74%	99.99%	1.703	1.948	0.053	0.082

Table 10 GLOBALLib Instances with non-zero Duality Gap (Part 1)

			% Duality Gap Closed				Time Taken (sec)			
Instance	**RLT**	**OPT**	**W1**	**W2**	**V1**	**V2**	**W1**	**W2**	**V1**	**V2**
house	-5230.54	-4500.00	77.96%	80.33%	0.00%	86.93%	0.726	0.974	0.435	12.873
hydro	4019717.93	4366944.16	100.00%	100.00%	100.00%	100.00%	0.211	0.252	8.354	20.668
mathopt1	-912909.01	1.00	100.00%	100.00%	100.00%	100.00%	0.015	0.023	1.727	2.448
mathopt2	-11289.00	0.00	100.00%	100.00%	100.00%	100.00%	0.039	0.033	0.351	0.229
meanvar	0.00	5.24	100.00%	100.00%	100.00%	100.00%	0.009	0.008	0.179	0.276
nemhaus	0.00	31.00	100.00%	100.00%	53.97%	100.00%	0.025	0.025	0.836	0.198
prob05	0.32	0.74	61.73%	61.79%	0.00%	99.78%	0.134	0.099	0.007	0.165
prob06	1.00	1.18	100.00%	100.00%	100.00%	100.00%	0.036	0.038	0.023	0.024
prob09	-100.00	0.00	100.00%	100.00%	100.00%	99.99%	0.070	0.079	0.582	0.885
process	-2756.59	-1161.34	85.00%	84.88%	7.68%	88.05%	13.965	11.318	6.379	3620.085
qp1	-1.43	0.00	100.00%	100.00%	85.76%	89.12%	0.034	0.035	3659.085	3897.521
qp2	-1.43	0.00	100.00%	100.00%	86.13%	89.15%	0.035	0.034	3643.188	4047.592
rbrock	-659984.01	-5.67	100.00%	100.00%	100.00%	100.00%	0.012	0.010	0.353	3.194
st_bpaf1a	-46.01	-45.38	0.00%	0.00%	0.00%	81.73%	0.080	0.114	0.049	0.894
st_bpaf1b	-43.13	-42.96	-0.01%	-0.01%	0.00%	90.73%	0.075	0.114	0.047	3.299
st_bpv2	-11.25	-8.00	99.97%	99.97%	0.00%	99.99%	0.022	0.025	0.033	0.029
st_bsj2	-0.63	1.00	95.76%	99.95%	0.00%	99.98%	0.207	0.083	0.009	1.974
st_bsj3	-86768.55	-86768.55	-0.03%	-0.03%	0.00%	0.00%	5.533	5.136	0.012	0.011
st_bsj4	-72700.05	-70262.05	93.34%	93.34%	0.00%	99.86%	1.570	1.579	0.014	1.715
st_e02	171.42	201.16	91.82%	95.88%	0.00%	99.88%	0.053	0.088	0.008	0.095
st_e03	-2381.89	-1161.34	82.12%	91.95%	29.58%	91.63%	15.687	1326.453	715.006	3639.297
st_e05	3826.39	7049.25	9.80%	9.80%	0.00%	50.43%	0.209	0.133	0.194	16.217
st_e06	0.00	0.16	0.00%	0.00%	0.00%	0.00%	0.117	0.260	0.215	0.726
st_e07	-500.00	-400.00	85.66%	85.67%	0.00%	99.97%	0.319	0.809	0.042	0.350
st_e08	0.31	0.74	61.76%	61.82%	0.00%	99.81%	0.134	0.095	0.008	0.208
st_e09	-0.75	-0.50	91.77%	91.77%	0.00%	92.58%	0.048	0.067	0.012	0.014
st_e10	-29.00	-16.74	100.00%	100.00%	100.00%	100.00%	0.028	0.030	0.036	0.045
st_e18	-3.00	-2.83	100.00%	100.00%	100.00%	100.00%	0.015	0.028	0.015	0.018
st_e19	-879.75	-86.42	93.51%	93.51%	93.50%	95.21%	0.118	0.146	0.373	0.613
st_e20	-0.85	-0.39	66.29%	65.42%	0.00%	76.38%	7.402	9.537	7.409	3610.271
st_e23	-3.00	-1.08	96.42%	96.42%	0.00%	98.40%	0.924	0.943	0.011	0.087
st_e24	0.00	3.00	66.58%	66.58%	0.00%	99.81%	0.022	0.024	0.007	0.501
st_e25	0.25	0.89	100.00%	100.00%	87.20%	100.00%	0.017	0.015	0.312	0.161
st_e26	-513.00	-185.78	99.99%	99.99%	0.00%	99.96%	0.032	0.034	0.006	0.036
st_e28	-30802.76	-30665.54	26.51%	59.00%	49.74%	99.99%	1.721	1.934	0.051	0.088
st_e30	-3.00	-1.58	0.00%	0.00%	0.00%	0.00%	0.286	0.981	0.014	0.035
st_e33	-500.00	-400.00	85.79%	85.78%	0.00%	99.94%	0.172	0.269	0.047	0.457
st_fp1	-18.90	-17.00	99.96%	99.96%	0.00%	72.62%	1.263	1.315	0.009	658.824
st_fp5	-269.45	-268.01	77.09%	99.95%	0.00%	99.98%	0.052	0.104	0.018	0.175
st_fp6	-44.40	-39.00	99.97%	99.96%	0.00%	99.92%	0.376	0.286	0.025	3603.767
st_fp7a	-435.52	-354.75	63.32%	98.55%	0.00%	45.13%	1.329	49.026	0.151	806.493
st_fp7b	-715.52	-634.75	63.32%	98.59%	0.00%	22.06%	1.285	45.861	0.153	11.941
st_fp7c	-10310.47	-8695.01	63.32%	98.35%	0.00%	44.26%	1.317	44.759	0.181	3621.180

Table 11 GLOBALLib Instances with non-zero Duality Gap (Part 2)

			% Duality Gap Closed				Time Taken (sec)			
Instance	**RLT**	**OPT**	**W1**	**W2**	**V1**	**V2**	**W1**	**W2**	**V1**	**V2**
st_fp7d	-195.52	-114.75	63.32%	96.95%	0.00%	50.03%	1.378	24.630	0.111	3627.749
st_fp8	7219.50	15639.00	5.06%	6.85%	0.00%	0.83%	5.516	6.583	0.331	4.911
st_glmp_fp2	7.07	7.34	0.00%	0.00%	0.00%	45.70%	0.028	0.028	0.009	0.732
st_glmp_kk92	-13.35	-12.00	100.00%	100.00%	0.00%	99.98%	0.027	0.027	0.023	0.038
st_glmp_kky	-3.00	-2.50	0.00%	99.62%	0.00%	99.80%	0.052	0.049	0.011	0.133
st_glmp_ss1	-38.67	-24.57	73.96%	73.96%	0.00%	89.30%	0.042	0.036	0.031	0.556
st_ht	-2.80	-1.60	91.58%	99.88%	0.00%	99.81%	0.083	0.066	0.006	0.142
st_iqpbk1	-1722.38	-621.49	99.13%	99.97%	97.99%	99.86%	0.263	0.179	3.825	5.086
st_iqpbk2	-3441.95	-1195.23	99.16%	99.99%	97.93%	100.00%	0.303	0.178	2.515	31.614
st_jcbpaf2	-945.45	-794.86	75.21%	85.77%	0.00%	99.47%	0.391	0.453	2.650	3622.733
st_jcbpafex	-3.00	-1.08	96.42%	96.42%	0.00%	98.40%	0.929	0.859	0.012	0.085
st_kr	-104.00	-85.00	63.09%	99.95%	0.00%	99.93%	0.030	0.047	0.008	0.090
st_m1	-505191.34	-461356.94	83.08%	87.36%	0.00%	99.96%	0.300	1.365	0.222	368.618
st_m2	-938513.68	-856648.82	66.40%	66.40%	0.00%	70.19%	16.190	21.288	1.226	3641.449
st_pan1	-5.69	-5.28	99.89%	99.86%	0.00%	99.72%	0.098	0.090	0.007	0.926
st_pan2	-19.40	-17.00	99.96%	99.96%	0.00%	68.54%	9.898	9.683	0.009	3038.430
st_ph1	-243.81	-230.12	99.92%	99.93%	0.00%	99.98%	0.039	0.063	0.011	0.225
st_ph11	-11.75	-11.28	53.14%	53.14%	0.00%	99.46%	0.044	0.052	0.007	0.910
st_ph12	-23.50	-22.63	56.90%	56.90%	0.00%	99.49%	0.093	0.088	0.006	0.353
st_ph13	-11.75	-11.28	53.17%	53.17%	0.00%	99.38%	0.050	0.051	0.009	0.751
st_ph14	-231.00	-229.72	78.04%	78.04%	0.00%	99.85%	0.042	0.044	0.010	0.051
st_ph15	-434.73	-392.70	58.20%	99.90%	0.00%	99.83%	0.040	0.050	0.009	0.476
st_ph2	-1064.50	-1028.12	99.94%	99.94%	0.00%	99.98%	0.039	0.062	0.014	0.159
st_ph20	-178.00	-158.00	89.96%	99.97%	0.00%	99.98%	0.029	0.046	0.007	0.036
st_ph3	-447.85	-420.23	59.08%	99.97%	0.00%	99.98%	0.033	0.042	0.011	0.031
st_phex	-104.00	-85.00	63.09%	99.95%	0.00%	99.96%	0.028	0.052	0.007	0.088
st_qpc-m0	-6.00	-5.00	99.91%	99.94%	0.00%	99.96%	0.020	0.024	0.007	0.015
st_qpc-m1	-612.27	-473.78	98.52%	100.00%	0.00%	99.99%	0.063	0.095	0.009	0.223
st_qpc-m3a	-725.05	-382.70	99.69%	99.99%	0.00%	98.10%	0.069	0.119	0.025	3615.442
st_qpc-m3b	-24.68	0.00	99.06%	99.99%	0.00%	100.00%	0.805	0.231	0.021	0.566
st_cqpf	-5002.00	-2.75	100.00%	100.00%	-	-	0.011	0.011	-	-
st_cqpjk2	-18.00	-12.50	100.00%	100.00%	-	-	0.014	0.010	-	-
st_qpk1	-11.00	-3.00	99.97%	99.99%	0.00%	99.98%	0.062	0.032	0.007	0.110
st_qpk2	-21.00	-12.25	60.03%	68.03%	0.00%	71.34%	63.905	116.628	0.025	3599.788
st_qpk3	-66.00	-36.00	32.24%	32.64%	0.00%	33.53%	258.712	399.522	0.077	3621.930
st_rv1	-64.24	-59.94	57.84%	76.36%	0.00%	96.19%	0.178	0.247	0.023	3607.723
st_rv2	-73.00	-64.48	85.50%	85.50%	0.00%	88.79%	1.067	1.473	0.079	3601.528
st_rv3	-38.52	-35.76	80.88%	81.43%	0.00%	40.40%	1.219	1.254	0.108	112.028
st_rv7	-148.98	-138.19	77.33%	90.38%	0.00%	45.43%	1.615	5.018	0.269	3640.861
st_rv8	-143.58	-132.66	81.02%	86.31%	0.00%	29.90%	4.999	11.664	0.663	3696.452
st_rv9	-134.91	-120.12	83.94%	85.90%	0.00%	20.56%	84.413	102.962	1.019	3920.213
st_z	-0.97	0.00	91.14%	99.93%	0.00%	99.96%	0.122	0.102	0.009	2.749

Table 12 GLOBALLib Instances with non-zero Duality Gap (Part 3)

		Time Taken (sec)		Max Infeasibility	
Instance	**OPT**	**W1**	**W2**	**W1**	**W2**
ex14_1_2	0.00	5.016	87.542	0.163	0.333
ex14_1_6	0.00	0.571	0.819	0.306	0.266
ex2_1_2	-213.00	0.012	0.026	0.000	0.000
ex2_1_3	-15.00	0.037	0.048	0.000	0.000
ex2_1_4	-11.00	0.008	0.010	0.000	0.000
st_bpk1	-13.00	0.015	0.017	0.000	0.000
st_bpk2	-13.00	0.013	0.016	0.000	0.000
st_bpv1	10.00	0.010	0.015	0.000	0.000
st_e01	-6.67	0.025	0.027	0.049	0.049
st_e17	0.00	0.010	0.010	0.000	0.000
st_e34	0.02	0.059	0.105	0.094	0.058
st_e42	18.78	0.061	0.079	0.069	0.000
st_fp2	-213.00	0.011	0.026	0.000	0.000
st_fp3	-15.00	0.039	0.051	0.000	0.000
st_fp4	-11.00	0.013	0.010	0.000	0.000
st_glmp_fp1	10.00	0.011	0.011	0.000	0.000
st_glmp_fp3	-12.00	0.013	0.015	0.000	0.000
st_glmp_kk90	3.00	0.024	0.027	0.005	0.005
st_glmp_ss2	3.00	0.029	0.029	0.041	0.041
st_ph10	-10.50	0.008	0.013	0.000	0.000
st_qpc-m3c	0.00	0.049	0.077	0.038	0.000
st_qpc-m4	0.00	0.090	0.229	0.056	0.034
st_robot	0.00	0.371	0.752	0.335	0.247

Table 13 GLOBALLib Instances with zero Duality Gap

			% Duality Gap Closed				Time Taken (sec)	
Instance	**RLT**	**OPT**	**W1**	**W2**	**W1+LCD**	**W2+LCD**	**W1+LCD**	**W2+LCD**
ex9_1_4	-63.00	-37.00	0.00%	0.00%	100.00%	99.99%	0.749	0.729
ex9_2_1	-16.00	17.00	68.14%	68.74%	86.36%	86.36%	0.961	0.897
ex9_2_2	-50.00	100.00	89.62%	93.95%	100.00%	100.00%	1.656	0.802
ex9_2_3	-30.00	0.00	0.00%	0.00%	99.96%	99.97%	0.265	1.835
ex9_2_4	-396.00	0.50	99.87%	99.87%	100.00%	100.00%	0.116	0.140
ex9_2_6	-406.00	-1.00	99.88%	99.88%	99.88%	99.88%	0.519	0.382
ex9_2_7	-9.00	17.00	59.30%	61.34%	82.68%	82.67%	0.965	0.938
ex9_2_8	0.50	1.50	100.00%	100.00%	100.00%	100.00%	0.070	0.087

Table 14 GLOBALLib Instances with Linear Complementarity Constraints

			% Duality Gap Closed		Time Taken (sec)	
Instance	**RLT**	**OPT**	**W1-Dsj**	**W2-Dsj**	**W1-Dsj**	**W2-Dsj**
alkyl	-2.76	-1.77	41.82%	63.77%	0.206	1.554
circle	0.00	4.57	75.66%	90.17%	0.029	0.172
dispatch	3101.28	3155.29	100.00%	100.00%	0.010	0.026
ex2_1_1	-18.90	-17.00	0.00%	0.00%	0.012	0.024
ex2_1_10	39668.06	49318.02	40.60%	62.93%	0.029	0.131
ex2_1_5	-269.45	-268.01	26.00%	99.95%	0.022	0.086
ex2_1_6	-44.40	-39.00	64.08%	99.34%	0.020	0.165
ex2_1_7	-6031.90	-4150.41	11.26%	48.58%	0.041	1.022
ex2_1_8	-82460.00	15639.00	98.78%	99.47%	0.096	1.096
ex2_1_9	-2.20	-0.38	84.39%	92.96%	0.015	0.060
ex3_1_1	2533.20	7049.25	0.10%	0.11%	0.089	0.176
ex3_1_2	-30802.76	-30665.54	11.19%	47.06%	5.723	7.181
ex3_1_3	-440.00	-310.00	97.69%	100.00%	0.017	0.078
ex3_1_4	-6.00	-4.00	0.00%	15.33%	0.011	0.041
ex4_1_1	-173688.80	-7.49	99.41%	99.88%	0.074	0.152
ex4_1_3	-7999.46	-443.67	71.40%	77.14%	0.047	0.181
ex4_1_4	-200.00	0.00	50.00%	50.00%	0.035	0.105
ex4_1_6	-24075.00	7.00	62.08%	62.08%	0.022	0.108
ex4_1_7	-206.25	-7.50	68.90%	85.46%	0.041	0.094
ex4_1_8	-29.00	-16.74	100.00%	100.00%	0.010	0.019
ex4_1_9	-6.99	-5.51	0.00%	3.68%	0.035	0.057
ex5_2_2_case1	-599.90	-400.00	-0.22%	-0.22%	0.166	0.116
ex5_2_2_case2	-1200.00	-600.00	-0.05%	-0.05%	0.049	0.126
ex5_2_2_case3	-875.00	-750.00	-0.02%	-0.01%	0.036	0.104
ex5_2_4	-2933.33	-450.00	39.28%	67.60%	0.043	0.083
ex5_3_2	1.00	1.86	0.00%	0.00%	0.116	1.549
ex5_4_2	2598.25	7512.23	0.11%	0.26%	0.070	0.068
ex7_3_2	0.00	1.09	0.00%	0.00%	0.020	0.130
ex8_1_4	-13.00	0.00	51.54%	51.54%	0.019	0.100
ex8_1_5	-3.33	0.00	0.00%	0.00%	0.084	0.226
ex8_1_7	-757.58	0.03	48.54%	85.58%	0.056	0.240
ex8_1_8	-0.85	-0.39	30.01%	55.36%	0.034	0.219
ex8_4_1	-5.00	0.62	88.99%	89.18%	0.041	1.022
ex8_4_2	-5.00	0.49	91.16%	91.35%	0.039	1.052
ex9_1_4	-63.00	-37.00	0.00%	0.00%	0.021	0.128
ex9_2_1	-16.00	17.00	54.55%	64.97%	0.022	0.122
ex9_2_2	-50.00	100.00	81.09%	93.95%	0.022	0.166
ex9_2_3	-30.00	0.00	0.00%	0.00%	0.018	0.272
ex9_2_4	-396.00	0.50	99.87%	99.87%	0.013	0.038
ex9_2_6	-406.00	-1.00	99.88%	99.88%	0.022	0.188
ex9_2_7	-9.00	17.00	42.31%	55.54%	0.023	0.109
ex9_2_8	0.50	1.50	100.00%	100.00%	0.013	0.032
himmel11	-30802.76	-30665.54	11.19%	44.77%	5.330	8.874

Table 15 Marginal Value of Disjunctive Cuts (Part 1)

			% Duality Gap Closed		Time Taken (sec)	
Instance	**RLT**	**OPT**	**W1-Dsj**	**W2-Dsj**	**W1-Dsj**	**W2-Dsj**
house	-5230.54	-4500.00	56.07%	79.63%	0.023	0.164
hydro	4019717.93	4366944.16	100.00%	100.00%	0.029	0.061
mathopt1	-912909.01	1.00	100.00%	100.00%	0.009	0.014
mathopt2	-11289.00	0.00	100.00%	100.00%	0.014	0.028
meanvar	0.00	5.24	100.00%	100.00%	0.007	0.008
nemhaus	0.00	31.00	100.00%	100.00%	0.015	0.015
prob05	0.32	0.74	32.40%	52.66%	0.020	0.030
prob06	1.00	1.18	100.00%	100.00%	0.018	0.021
prob09	-100.00	0.00	100.00%	100.00%	0.022	0.043
process	-2756.59	-1161.34	52.33%	73.56%	2.764	3.288
qp1	-1.43	0.00	100.00%	100.00%	0.035	0.034
qp2	-1.43	0.00	100.00%	100.00%	0.036	0.036
rbrock	-659984.01	-5.67	100.00%	100.00%	0.009	0.007
st_bpaf1a	-46.01	-45.38	0.00%	0.00%	0.037	0.064
st_bpaf1b	-43.13	-42.96	-0.01%	-0.01%	0.025	0.066
st_bpv2	-11.25	-8.00	89.16%	89.16%	0.008	0.011
st_bsj2	-0.63	1.00	40.50%	83.30%	0.012	0.042
st_bsj3	-86768.55	-86768.55	-0.03%	-0.03%	5.429	5.564
st_bsj4	-72700.05	-70262.05	0.00%	0.00%	4.387	4.613
st_e02	171.42	201.16	41.79%	95.88%	0.013	0.053
st_e03	-2381.89	-1161.34	50.18%	72.27%	3.704	6.350
st_e05	3826.39	7049.25	2.70%	8.25%	0.089	0.118
st_e06	0.00	0.16	0.00%	0.00%	0.032	0.193
st_e07	-500.00	-400.00	0.00%	0.00%	0.086	0.564
st_e08	0.31	0.74	32.35%	52.61%	0.023	0.031
st_e09	-0.75	-0.50	75.87%	75.87%	0.017	0.021
st_e10	-29.00	-16.74	100.00%	100.00%	0.010	0.019
st_e18	-3.00	-2.83	100.00%	100.00%	0.008	0.018
st_e19	-879.75	-86.42	93.51%	93.51%	0.045	0.080
st_e20	-0.85	-0.39	30.01%	55.36%	0.037	0.228
st_e23	-3.00	-1.08	95.06%	95.06%	0.008	0.011
st_e24	0.00	3.00	0.00%	0.00%	0.009	0.010
st_e25	0.25	0.89	100.00%	100.00%	0.011	0.012
st_e26	-513.00	-185.78	96.14%	99.99%	0.013	0.022
st_e28	-30802.76	-30665.54	11.19%	44.77%	5.433	8.741
st_e30	-3.00	-1.58	0.00%	0.00%	0.076	0.411
st_e33	-500.00	-400.00	0.00%	0.00%	0.368	0.290
st_fp1	-18.90	-17.00	0.00%	0.00%	0.010	0.025
st_fp5	-269.45	-268.01	26.00%	99.95%	0.023	0.086
st_fp6	-44.40	-39.00	64.08%	99.34%	0.021	0.166
st_fp7a	-435.52	-354.75	0.00%	50.72%	0.043	1.021
st_fp7b	-715.52	-634.75	0.00%	50.72%	0.045	1.015
st_fp7c	-10310.47	-8695.01	0.00%	50.72%	0.040	1.031

Table 16 Marginal Value of Disjunctive Cuts (Part 2)

			% Duality Gap Closed		Time Taken (sec)	
Instance	**RLT**	**OPT**	**W1-Dsj**	**W2-Dsj**	**W1-Dsj**	**W2-Dsj**
st_fp7d	-195.52	-114.75	0.00%	50.72%	0.037	1.032
st_fp8	7219.50	15639.00	0.00%	4.04%	18.493	6.371
st_glmp_fp2	7.07	7.34	0.00%	0.00%	0.018	0.020
st_glmp_kk92	-13.35	-12.00	100.00%	100.00%	0.019	0.021
st_glmp_kky	-3.00	-2.50	0.00%	99.62%	0.024	0.038
st_glmp_ss1	-38.67	-24.57	54.28%	54.28%	0.016	0.017
st_ht	-2.80	-1.60	0.00%	99.88%	0.009	0.045
st_iqpbk1	-1722.38	-621.49	97.52%	99.92%	0.072	0.139
st_iqpbk2	-3441.95	-1195.23	97.41%	99.91%	0.083	0.147
st_jcbpaf2	-945.45	-794.86	18.93%	67.56%	0.029	0.106
st_jcbpafex	-3.00	-1.08	95.06%	95.06%	0.011	0.010
st_kr	-104.00	-85.00	0.00%	99.95%	0.008	0.034
st_m1	-505191.34	-461356.94	6.46%	86.08%	0.049	1.234
st_m2	-938513.68	-856648.82	0.00%	28.80%	58.782	21.975
st_pan1	-5.69	-5.28	0.00%	40.11%	0.010	0.053
st_pan2	-19.40	-17.00	0.00%	0.00%	0.010	0.029
st_ph1	-243.81	-230.12	0.00%	99.91%	0.017	0.051
st_ph11	-11.75	-11.28	0.00%	0.00%	0.012	0.017
st_ph12	-23.50	-22.63	0.00%	0.00%	0.013	0.017
st_ph13	-11.75	-11.28	0.00%	0.00%	0.012	0.017
st_ph14	-231.00	-229.72	0.00%	0.00%	0.010	0.018
st_ph15	-434.73	-392.70	0.00%	99.90%	0.017	0.035
st_ph2	-1064.50	-1028.12	0.00%	99.91%	0.019	0.059
st_ph20	-178.00	-158.00	75.00%	99.97%	0.014	0.034
st_ph3	-447.85	-420.23	0.00%	99.97%	0.017	0.037
st_phex	-104.00	-85.00	0.00%	99.95%	0.010	0.037
st_qpc-m0	-6.00	-5.00	0.00%	99.94%	0.011	0.019
st_qpc-m1	-612.27	-473.78	95.68%	100.00%	0.018	0.068
st_qpc-m3a	-725.05	-382.70	98.60%	99.99%	0.023	0.084
st_qpc-m3b	-24.68	0.00	84.21%	99.99%	0.097	0.156
st_cqpf	-5002.00	-2.75	100.00%	100.00%	0.009	0.012
st_cqpjk2	-18.00	-12.50	100.00%	100.00%	0.013	0.014
st_qpk1	-11.00	-3.00	83.33%	99.99%	0.016	0.033
st_qpk2	-21.00	-12.25	0.00%	44.15%	0.015	0.110
st_qpk3	-66.00	-36.00	0.00%	2.16%	0.017	0.331
st_rv1	-64.24	-59.94	0.00%	42.18%	0.023	0.133
st_rv2	-73.00	-64.48	0.00%	4.03%	0.034	0.453
st_rv3	-38.52	-35.76	0.00%	46.56%	0.035	0.528
st_rv7	-148.98	-138.19	0.00%	31.69%	0.058	1.338
st_rv8	-143.58	-132.66	0.00%	39.37%	0.081	2.607
st_rv9	-134.91	-120.12	0.00%	15.07%	0.114	4.527
st_z	-0.97	0.00	0.00%	71.94%	0.013	0.051

Table 17 Marginal Value of Disjunctive Cuts (Part 3)

			% Duality Gap Closed		Time Taken (sec)			%Time spent on Cut Generation		Time (sec) to solve last relaxation	
Instance	RLT	OPT	W3	W3-SDP	W3	W3-SDP	W3 (Adj)	W3	W3-SDP	W3	W3-SDP
spar020-100-1	-1066.00	-706.50	98.28%	96.83%	43.06	1.25	1.60	48.39%	0.16%	0.33	0.01
spar020-100-2	-1289.00	-856.50	94.61%	91.54%	2.49	0.84	1.27	26.28%	0.12%	0.05	0.01
spar020-100-3	-1168.50	-772.00	99.98%	99.92%	408.36	2.46	0.51	95.77%	0.24%	0.30	0.09
spar030-060-1	-1454.75	-706.00	93.84%	59.05%	13.40	2.09	8.39	61.99%	0.00%	0.09	0.01
spar030-060-2	-1699.50	-1377.17	97.35%	96.86%	50.79	7.29	7.54	25.01%	0.01%	0.62	0.18
spar030-060-3	-2047.00	-1293.50	95.62%	82.13%	33.92	14.38	13.46	26.09%	0.13%	0.30	0.22
spar030-070-1	-1569.00	-654.00	89.88%	51.41%	12.33	1.74	6.90	68.98%	0.06%	0.07	0.01
spar030-070-2	-1940.25	-1313.00	98.51%	89.02%	188.12	7.97	14.49	72.26%	0.03%	0.71	0.02
spar030-070-3	-2302.75	-1657.40	96.07%	95.14%	31.57	8.23	10.04	17.78%	0.00%	0.38	0.02
spar030-080-1	-2107.50	-952.73	95.04%	61.27%	23.57	2.29	11.45	64.55%	0.04%	0.16	0.02
spar030-080-2	-2178.25	-1597.00	100.00%	94.54%	226.60	7.82	9.87	81.04%	0.01%	0.90	0.02
spar030-080-3	-2403.50	-1809.78	99.20%	98.80%	339.41	8.40	3.32	85.13%	0.21%	0.87	0.23
spar030-090-1	-2423.50	-1296.50	99.21%	76.19%	53.39	2.52	11.33	50.27%	0.08%	0.37	0.01
spar030-090-2	-2667.00	-1466.84	98.56%	77.88%	56.98	4.45	11.20	50.28%	0.02%	0.38	0.01
spar030-090-3	-2538.25	-1494.00	99.88%	84.76%	565.88	4.59	8.09	91.48%	0.02%	0.82	0.01
spar030-100-1	-2602.00	-1227.13	98.38%	77.42%	30.28	5.74	10.52	52.81%	0.00%	0.23	0.02
spar030-100-2	-2729.25	-1260.50	96.93%	67.57%	18.85	3.81	9.76	58.86%	0.05%	0.12	0.05
spar030-100-3	-2751.75	-1511.05	97.16%	84.89%	56.21	13.70	13.62	30.33%	0.11%	0.41	0.22
spar040-030-1	-1088.00	-839.50	97.64%	46.83%	117.60	39.28	49.76	30.31%	0.13%	1.07	0.63
spar040-030-2	-1635.00	-1429.00	91.60%	28.62%	68.46	30.06	41.16	31.84%	0.08%	0.57	0.54
spar040-030-3	-1303.25	-1086.00	93.04%	21.79%	104.80	32.99	48.74	27.50%	0.11%	1.01	0.47
spar040-040-1	-1606.25	-837.00	87.85%	37.96%	43.71	11.69	28.13	56.95%	0.02%	0.24	0.02
spar040-040-2	-1920.75	-1428.00	99.61%	65.44%	114.57	14.68	33.10	57.04%	0.01%	0.50	0.03
spar040-040-3	-2039.75	-1173.50	92.94%	40.16%	35.77	5.81	20.40	69.33%	0.03%	0.19	0.03
spar040-050-1	-2146.25	-1154.50	93.71%	51.37%	43.86	15.28	30.26	56.57%	0.01%	0.17	0.03
spar040-050-2	-2357.25	-1430.98	95.17%	58.51%	54.14	30.01	28.08	48.71%	0.08%	0.27	0.52
spar040-050-3	-2616.00	-1653.63	94.81%	45.59%	44.05	6.83	23.83	64.04%	0.03%	0.26	0.03
spar040-060-1	-2872.00	-1322.67	93.47%	53.73%	46.67	16.47	28.01	59.90%	0.01%	0.16	0.03
spar040-060-2	-2917.50	-2004.23	96.20%	74.75%	80.14	17.98	32.18	32.45%	0.01%	0.55	0.03
spar040-060-3	-3434.00	-2454.50	99.18%	89.64%	134.80	65.28	25.19	34.00%	0.13%	1.19	0.75
spar040-070-1	-3144.00	-1605.00	98.85%	75.26%	101.61	38.09	29.88	58.41%	0.12%	0.42	0.57
spar040-070-2	-3369.25	-1867.50	98.56%	68.15%	94.96	10.26	28.65	52.82%	0.02%	0.48	0.03
spar040-070-3	-3760.25	-2436.50	97.83%	75.76%	112.96	28.45	33.15	53.72%	0.10%	0.49	0.48
spar040-080-1	-3846.50	-1838.50	98.43%	59.27%	134.03	5.18	27.67	76.06%	0.04%	0.46	0.02
spar040-080-2	-3833.00	-1952.50	98.26%	69.92%	47.06	8.05	22.08	59.83%	0.02%	0.32	0.02
spar040-080-3	-4361.50	-2545.50	97.98%	79.89%	83.80	7.88	20.05	41.85%	0.03%	0.50	0.03
spar040-090-1	-4376.75	-2135.50	98.22%	69.95%	103.96	6.00	22.36	64.65%	0.03%	0.49	0.02
spar040-090-2	-4357.75	-2113.00	98.04%	72.71%	83.69	12.91	29.92	61.39%	0.02%	0.37	0.03
spar040-090-3	-4516.75	-2535.00	99.00%	77.84%	81.20	8.78	22.64	47.57%	0.02%	0.51	0.02
spar040-100-1	-5009.75	-2476.38	98.72%	78.01%	81.56	4.64	13.75	53.00%	0.04%	0.50	0.03
spar040-100-2	-4902.75	-2102.50	97.93%	66.22%	121.76	4.87	24.09	74.29%	0.04%	0.42	0.02
spar040-100-3	-5075.75	-1866.07	95.87%	62.91%	40.16	4.16	18.58	78.37%	0.05%	0.18	0.02
spar050-030-1	-1858.25	-1324.50	96.40%	26.47%	165.74	99.13	76.19	51.02%	0.11%	0.87	1.01
spar050-030-2	-2334.00	-1668.00	90.74%	19.51%	79.42	17.41	48.81	58.12%	0.02%	0.34	0.08
spar050-030-3	-2107.25	-1453.61	91.45%	11.38%	121.65	14.01	61.02	65.49%	0.01%	0.53	0.04

Table 18 Box Constrained QPs from [22] (Part 1)

			% Duality Gap Closed		Time Taken (sec)			%Time spent on Cut Generation		Time (sec) to solve last relaxation	
Instance	RLT	OPT	W3	W3-SDP	W3	W3-SDP	W3 (Adj)	W3	W3-SDP	W3	W3-SDP
spar050-040-1	-2632.00	-1411.00	97.23%	46.45%	177.96	21.67	56.43	68.17%	0.01%	0.59	0.04
spar050-040-2	-2923.25	-1745.76	94.06%	41.74%	85.63	36.51	55.35	54.18%	0.04%	0.31	0.75
spar050-040-3	-3273.50	-2094.50	97.53%	46.70%	180.96	28.88	56.47	63.83%	0.02%	0.58	0.56
spar050-050-1	-3536.00	-1198.41	87.88%	32.46%	50.22	8.76	28.95	77.51%	0.05%	0.13	0.18
spar050-050-2	-3500.50	-1776.00	93.13%	44.26%	67.20	25.35	51.30	56.62%	0.02%	0.25	0.05
spar050-050-3	-4119.75	-2106.10	95.01%	50.16%	93.62	8.80	39.93	79.73%	0.03%	0.31	0.05
spar060-020-1	-1757.25	-1212.00	91.00%	0.00%	163.42	122.87	100.74	56.61%	0.10%	0.72	2.17
spar060-020-2	-2238.25	-1925.50	90.22%	0.00%	226.11	127.47	141.77	46.29%	0.11%	1.55	1.88
spar060-020-3	-2098.75	-1483.00	85.78%	0.00%	121.83	111.07	86.28	53.51%	0.12%	0.54	1.65
spar070-025-1	-3832.75	-2538.91	92.61%	9.73%	249.97	36.42	143.35	74.29%	0.01%	0.77	0.13
spar070-025-2	-3248.00	-1888.00	89.79%	0.00%	191.12	202.98	107.47	71.13%	0.11%	0.81	2.42
spar070-025-3	-4167.25	-2812.28	90.68%	8.63%	214.40	26.02	123.93	72.44%	0.02%	0.70	0.40
spar070-050-1	-7210.75	-3252.50	94.40%	42.80%	240.93	35.55	131.39	75.21%	0.01%	0.48	0.12
spar070-050-2	-6620.00	-3296.00	95.77%	40.78%	283.03	28.63	130.32	80.06%	0.02%	0.57	0.08
spar070-050-3	-7522.00	-4306.50	99.36%	53.54%	693.28	33.70	125.71	83.82%	0.01%	1.10	0.08
spar070-075-1	-11647.75	-4655.50	96.90%	53.67%	365.50	23.91	109.01	85.42%	0.02%	0.60	0.12
spar070-075-2	-10884.75	-3865.15	95.57%	52.30%	293.31	23.71	92.63	84.77%	0.03%	0.49	0.12
spar070-075-3	-11262.25	-4329.40	96.18%	53.10%	342.92	22.02	104.20	87.70%	0.02%	0.62	0.10
spar080-025-1	-4840.75	-3157.00	93.91%	3.57%	524.07	45.61	230.53	77.38%	0.01%	1.34	0.13
spar080-025-2	-4378.50	-2312.34	88.14%	2.95%	257.62	42.15	151.85	77.35%	0.01%	0.57	0.17
spar080-025-3	-5130.25	-3090.88	91.59%	8.99%	420.61	43.34	159.31	76.37%	0.02%	1.08	0.82
spar080-050-1	-9783.25	-3448.10	92.65%	38.88%	355.97	36.43	121.62	82.84%	0.02%	0.67	0.16
spar080-050-2	-9270.00	-4449.20	97.50%	44.21%	892.96	50.13	202.59	83.53%	0.01%	2.03	0.10
spar080-050-3	-10029.75	-4886.00	95.58%	43.70%	435.41	34.77	152.68	84.02%	0.01%	0.80	0.16
spar080-075-1	-15250.75	-5896.00	96.93%	54.91%	387.48	37.72	136.06	84.44%	0.02%	0.64	0.16
spar080-075-2	-14246.50	-5341.00	96.95%	56.24%	450.96	67.66	179.97	79.45%	0.01%	0.65	0.13
spar080-075-3	-14961.50	-5980.50	96.11%	54.74%	416.32	54.59	145.80	81.23%	0.01%	0.83	0.15
spar090-025-1	-6171.50	-3372.50	90.12%	10.54%	408.73	65.24	237.60	77.36%	0.01%	0.78	0.28
spar090-025-2	-6015.00	-3500.29	89.45%	7.01%	444.30	85.72	244.73	73.94%	0.01%	0.85	0.16
spar090-025-3	-6698.25	-4299.00	90.57%	5.73%	446.74	95.33	255.53	73.44%	0.02%	0.85	2.11
spar090-050-1	-12584.00	-5152.00	95.02%	42.95%	506.72	95.66	233.44	76.98%	0.01%	0.78	1.96
spar090-050-2	-11920.50	-5386.50	96.61%	44.48%	514.05	64.69	184.63	79.57%	0.01%	1.47	0.17
spar090-050-3	-12514.00	-6151.00	95.90%	42.69%	991.04	60.29	294.92	83.28%	0.01%	1.51	0.12
spar090-075-1	-19054.25	-6267.45	95.66%	49.15%	462.16	51.91	214.79	83.63%	0.01%	0.87	0.28
spar090-075-2	-18245.50	-5647.50	95.92%	49.61%	784.59	46.98	207.58	88.25%	0.01%	1.02	0.16
spar090-075-3	-18929.50	-6450.00	96.11%	50.13%	602.44	56.31	220.36	85.24%	0.01%	0.78	0.20
spar100-025-1	-7660.75	-4027.50	92.36%	12.27%	670.15	93.72	385.64	78.64%	0.01%	1.14	0.22
spar100-025-2	-7338.50	-3892.56	92.16%	8.17%	538.03	77.98	321.79	77.49%	0.01%	1.52	0.32
spar100-025-3	-7942.25	-4453.50	93.26%	9.83%	656.59	75.49	299.23	80.93%	0.01%	1.25	0.13
spar100-050-1	-15415.75	-5490.00	93.62%	38.34%	757.14	88.35	286.59	83.57%	0.01%	1.07	0.26
spar100-050-2	-14920.50	-5866.00	94.13%	39.65%	929.91	89.45	288.09	83.81%	0.01%	1.26	0.19
spar100-050-3	-15564.25	-6485.00	95.81%	39.88%	747.46	99.90	279.41	84.99%	0.01%	0.82	0.28
spar100-075-1	-23387.50	-7384.20	95.84%	49.95%	1509.96	112.69	366.24	92.30%	0.23%	2.01	2.50
spar100-075-2	-22440.00	-6755.50	96.47%	51.80%	936.61	81.78	330.70	86.75%	0.01%	1.24	0.38
spar100-075-3	-23243.50	-7554.00	96.06%	51.71%	657.84	75.81	303.30	84.22%	0.01%	0.88	0.31

Table 19 Box Constrained QPs from [22] (Part 2)

	% Duality Gap Closed			Time Taken (sec)			Time to solve last relaxation (sec)
Instance	SDPLR	SDPA	W3	SDPLR	SDPA	W3	W3
spar020-100-1	100.00%	100.00%	98.28%	5.33	3.04	43.06	0.33
spar020-100-2	99.67%	99.67%	94.61%	56.37	3.39	2.49	0.05
spar020-100-3	100.00%	100.00%	99.98%	0.97	1.98	408.36	0.30
spar030-060-1	98.84%	98.84%	93.84%	39.61	22.39	13.40	0.09
spar030-060-2	100.00%	100.00%	97.35%	3.76	18.32	50.79	0.62
spar030-060-3	99.38%	99.38%	95.62%	115.31	26.04	33.92	0.30
spar030-070-1	97.81%	97.81%	89.88%	21.39	22.35	12.33	0.07
spar030-070-2	100.00%	100.00%	98.51%	5.39	20.21	188.12	0.71
spar030-070-3	99.98%	99.98%	96.07%	234.24	29.33	31.57	0.38
spar030-080-1	98.92%	98.92%	95.04%	17.92	23.21	23.57	0.16
spar030-080-2	100.00%	100.00%	100.00%	4.02	16.66	226.60	0.90
spar030-080-3	100.00%	100.00%	99.20%	3.57	17.82	339.41	0.87
spar030-090-1	100.00%	100.00%	99.21%	6.50	19.88	53.39	0.37
spar030-090-2	100.00%	100.00%	98.56%	6.33	19.84	56.98	0.38
spar030-090-3	100.00%	100.00%	99.88%	4.40	17.34	565.88	0.82
spar030-100-1	100.00%	100.00%	98.38%	7.51	21.68	30.28	0.23
spar030-100-2	99.96%	99.96%	96.93%	59.94	26.46	18.85	0.12
spar030-100-3	99.84%	99.84%	97.16%	243.30	28.16	56.21	0.41
spar040-030-1	100.00%	100.00%	97.64%	13.46	115.06	117.60	1.07
spar040-030-2	100.00%	100.00%	91.60%	30.20	123.14	68.46	0.57
spar040-030-3	100.00%	100.00%	93.04%	16.85	120.88	104.80	1.01
spar040-040-1	96.61%	96.61%	87.85%	114.75	138.58	43.71	0.24
spar040-040-2	100.00%	100.00%	99.61%	12.08	105.68	114.57	0.50
spar040-040-3	99.15%	99.15%	92.94%	110.19	133.30	35.77	0.19
spar040-050-1	99.40%	99.40%	93.71%	68.94	152.34	43.86	0.17
spar040-050-2	99.46%	99.46%	95.17%	452.93	157.83	54.14	0.27
spar040-050-3	100.00%	100.00%	94.81%	51.68	141.97	44.05	0.26
spar040-060-1	98.05%	98.05%	93.47%	179.84	132.21	46.67	0.16
spar040-060-2	100.00%	100.00%	96.20%	22.91	127.96	80.14	0.55
spar040-060-3	100.00%	100.00%	99.18%	10.30	106.97	134.80	1.19
spar040-070-1	100.00%	100.00%	98.85%	24.00	143.54	101.61	0.42
spar040-070-2	100.00%	100.00%	98.56%	15.24	116.59	94.96	0.48
spar040-070-3	100.00%	100.00%	97.83%	81.80	138.93	112.96	0.49
spar040-080-1	100.00%	100.00%	98.43%	27.91	124.43	134.03	0.46
spar040-080-2	100.00%	100.00%	98.26%	19.78	119.97	47.06	0.32
spar040-080-3	99.99%	99.99%	97.98%	433.29	150.91	83.80	0.50
spar040-090-1	100.00%	100.00%	98.22%	52.66	153.75	103.96	0.49
spar040-090-2	99.97%	99.97%	98.04%	515.73	155.39	83.69	0.37
spar040-090-3	100.00%	100.00%	99.00%	17.56	114.04	81.20	0.51
spar040-100-1	100.00%	100.00%	98.72%	20.90	124.44	81.56	0.50
spar040-100-2	99.86%	99.86%	97.93%	131.30	147.47	121.76	0.42
spar040-100-3	98.69%	98.69%	95.87%	65.99	124.30	40.16	0.18
spar050-030-1	100.00%	100.00%	96.40%	41.72	458.44	165.74	0.87
spar050-030-2	99.50%	99.50%	90.74%	612.74	560.64	79.42	0.34

Table 20 Comparison with SDP Solvers (Part 1)

	% Duality Gap Closed			Time Taken (sec)			Time to solve last relaxation (sec)
Instance	**SDPLR**	**SDPA**	**W3**	**SDPLR**	**SDPA**	**W3**	**W3**
spar050-030-3	99.81%	99.81%	91.45%	477.50	589.17	121.65	0.53
spar050-040-1	100.00%	100.00%	97.23%	85.40	489.70	177.96	0.59
spar050-040-2	99.69%	99.69%	94.06%	416.80	572.74	85.63	0.31
spar050-040-3	100.00%	100.00%	97.53%	75.24	469.31	180.96	0.58
spar050-050-1	95.56%	95.56%	87.88%	170.32	438.77	50.22	0.13
spar050-050-2	99.21%	99.21%	93.13%	926.15	535.88	67.20	0.25
spar050-050-3	99.21%	99.21%	95.01%	404.45	524.92	93.62	0.31
spar060-020-1	100.00%	100.00%	91.00%	141.85	1400.78	163.42	0.72
spar060-020-2	100.00%	100.00%	90.22%	88.05	1150.06	226.11	1.55
spar060-020-3	98.69%	98.69%	85.78%	532.45	1408.32	121.83	0.54
spar070-025-1	99.54%	99.54%	92.61%	3600.75	3721.34	249.97	0.77
spar070-025-2	98.47%	98.46%	89.79%	1234.89	3420.19	191.12	0.81
spar070-025-3	98.97%	98.96%	90.68%	1646.93	3453.45	214.40	0.70
spar070-050-1	99.35%	99.35%	94.40%	2030.95	3465.12	240.93	0.48
spar070-050-2	99.87%	99.87%	95.77%	2193.35	3606.39	283.03	0.57
spar070-050-3	100.00%	100.00%	99.36%	133.07	2769.98	693.28	1.10
spar070-075-1	99.79%	99.79%	96.90%	1698.81	3531.73	365.50	0.60
spar070-075-2	98.85%	98.85%	95.57%	1164.56	3141.29	293.31	0.49
spar070-075-3	99.29%	99.29%	96.18%	1138.29	3180.22	342.92	0.62
spar080-025-1	100.00%	100.00%	93.91%	1020.64	8084.12	524.07	1.34
spar080-025-2	98.45%	98.44%	88.14%	1313.19	6618.79	257.62	0.57
spar080-025-3	99.36%	99.36%	91.59%	2341.98	7321.55	420.61	1.08
spar080-050-1	97.85%	97.85%	92.65%	965.18	6655.71	355.97	0.67
spar080-050-2	99.96%	99.96%	97.50%	3130.57	8285.12	892.96	2.03
spar080-050-3	99.33%	99.33%	95.58%	3839.60	8228.93	435.41	0.80
spar080-075-1	99.70%	99.70%	96.93%	1948.99	8200.16	387.48	0.64
spar080-075-2	99.46%	99.46%	96.95%	2537.80	7550.58	450.96	0.65
spar080-075-3	99.35%	99.35%	96.11%	5413.02	7333.87	416.32	0.83
spar090-025-1	97.83%	97.83%	90.12%	6793.35	13392.41	408.73	0.78
spar090-025-2	98.05%	98.05%	89.45%	2913.19	13823.48	444.30	0.85
spar090-025-3	98.23%	98.23%	90.57%	4514.18	14617.19	446.74	0.85
spar090-050-1	99.03%	99.03%	95.02%	4724.79	13657.86	506.72	0.78
spar090-050-2	100.00%	100.00%	96.61%	7049.49	17048.98	514.05	1.47
spar090-050-3	99.35%	99.35%	95.90%	5370.08	14548.34	991.04	1.51
spar090-075-1	98.94%	98.93%	95.66%	5166.41	13655.00	462.16	0.87
spar090-075-2	98.96%	98.96%	95.92%	2500.97	12838.46	784.59	1.02
spar090-075-3	99.35%	99.35%	96.11%	2403.62	12936.33	602.44	0.78
spar100-025-1	98.93%	98.93%	92.36%	5719.42	25368.87	670.15	1.14
spar100-025-2	99.09%	99.09%	92.16%	10185.65	26162.08	538.03	1.52
spar100-025-3	99.33%	99.33%	93.26%	5407.09	26139.26	656.59	1.25
spar100-050-1	98.17%	98.17%	93.62%	10139.57	23509.13	757.14	1.07
spar100-050-2	98.57%	98.57%	94.13%	5355.20	24356.26	929.91	1.26
spar100-050-3	99.39%	99.39%	95.81%	7281.26	26223.00	747.46	0.82
spar100-075-1	99.19%	99.19%	95.84%	9660.79	28604.12	1509.96	2.01
spar100-075-2	99.18%	99.18%	96.47%	6576.10	27198.30	936.61	1.24
spar100-075-3	99.19%	99.19%	96.06%	10295.88	27479.68	657.84	0.88

Table 21 Comparison with SDP Solvers (Part 2)