Alternative Mixed Integer Linear Programming Formulations for Globally Solving Standard Quadratic Programs

E. Alper Yıldırım

Koç University
Department of Industrial Engineering
Istanbul, Turkey

Computational Optimization in Action
University of Edinburgh

June 8, 2018

Joint work with Jacek Gondzio

Standard Quadratic Program

Definition

A **standard quadratic program** involves minimizing a (nonconvex) quadratic form (i.e., a homogeneous quadratic function) over the unit simplex.

Standard Quadratic Program

Definition

A **standard quadratic program** involves minimizing a (nonconvex) quadratic form (i.e., a homogeneous quadratic function) over the unit simplex.

$$(\mathrm{StQP}) \quad \nu(Q) = \min\{x^T Q x : x \in \Delta_n\},$$

where $\Delta_n \subset \mathbb{R}^n$ denotes the unit simplex given by

$$\Delta_n = \{x \in \mathbb{R}^n : e^T x = 1, \quad x \in \mathbb{R}^n_+\},$$

and

- $Q \in \mathcal{S}^n$, where $\mathcal{S}^n$ denotes the space of $n \times n$ real symmetric matrices,
- $x \in \mathbb{R}^n$,
- $e \in \mathbb{R}^n$ denotes the vector of all ones, and
- $\mathbb{R}^n_+$ denotes the nonnegative orthant in $\mathbb{R}^n$.

Basic Observations

- The term "standard quadratic program" was coined by Bomze. [Bomze, 1998]

Basic Observations

- The term "standard quadratic program" was coined by Bomze. [Bomze, 1998]
- Minimizing a quadratic form is not really restrictive:

$$x^T Qx + 2c^T x = x^T (Q + ce^T + ec^T)x, \quad \forall x \in \Delta_n.$$

Basic Observations

- The term "standard quadratic program" was coined by Bomze. [Bomze, 1998]
- Minimizing a quadratic form is not really restrictive:

$$x^T Qx + 2c^T x = x^T (Q + ce^T + ec^T)x, \quad \forall x \in \Delta_n.$$

- Therefore,

$$\min\{x^T Qx + 2c^T x : x \in \Delta_n\} = \min\{x^T Q'x : x \in \Delta_n\} = \nu(Q'),$$

where $Q' = Q + ce^T + ec^T \in \mathcal{S}^n$.

Basic Observations

- The term "standard quadratic program" was coined by Bomze. [Bomze, 1998]
- Minimizing a quadratic form is not really restrictive:

$$x^T Qx + 2c^T x = x^T (Q + ce^T + ec^T)x, \quad \forall x \in \Delta_n.$$

- Therefore,

$$\min\{x^T Qx + 2c^T x : x \in \Delta_n\} = \min\{x^T Q' x : x \in \Delta_n\} = \nu(Q'),$$

where $Q' = Q + ce^T + ec^T \in \mathcal{S}^n$.

- For any $\gamma \in \mathbb{R}$,

$$\begin{aligned} \nu(Q + \gamma ee^T) &= \min\{x^T (Q + \gamma ee^T)x : x \in \Delta_n\} \\ &= \gamma + \min\{x^T Qx : x \in \Delta_n\} \\ &= \nu(Q) + \gamma. \end{aligned}$$

Basic Observations

- The term "standard quadratic program" was coined by Bomze. [Bomze, 1998]
- Minimizing a quadratic form is not really restrictive:

$$x^T Qx + 2c^T x = x^T (Q + ce^T + ec^T)x, \quad \forall x \in \Delta_n.$$

- Therefore,

$$\min\{x^T Qx + 2c^T x : x \in \Delta_n\} = \min\{x^T Q' x : x \in \Delta_n\} = \nu(Q'),$$

where $Q' = Q + ce^T + ec^T \in \mathcal{S}^n$.

- For any $\gamma \in \mathbb{R}$,

$$\begin{aligned} \nu(Q + \gamma ee^T) &= \min\{x^T (Q + \gamma ee^T)x : x \in \Delta_n\} \\ &= \gamma + \min\{x^T Qx : x \in \Delta_n\} \\ &= \nu(Q) + \gamma. \end{aligned}$$

- Optimal solution of (StQP) is always attained.

Applications

- Portfolio optimization (e.g., [Markowitz, 1952])
- Quadratic resource allocation problem (e.g., [Ibaraki and Katoh, 1988])
- Maximum (weighted) stable set problem [Motzkin and Straus, 1965], [Gibbons et al., 1997]
- Social network analysis ([Bomze, 2018])
- Copositivity detection (a matrix $M \in \mathcal{S}^n$ is copositive iff $\nu(M) = \min\{x^T Mx : x \in \Delta_n\} \geq 0$)
- NP-hard in general

Motivation I

- In this talk, we are interested in solving (StQP) to global optimality.

Motivation I

- In this talk, we are interested in solving (StQP) to global optimality.
- Few specific global solution approaches for standard quadratic programs.

Motivation I

- In this talk, we are interested in solving (StQP) to global optimality.
- Few specific global solution approaches for standard quadratic programs.
- Adaptive simplicial partitioning [Bundfuss and Dür, 2009]

Motivation I

- In this talk, we are interested in solving (StQP) to global optimality.
- Few specific global solution approaches for standard quadratic programs.
- Adaptive simplicial partitioning [Bundfuss and Dür, 2009]
- DC-based branch-and-bound [Bomze, 2002]; clique-based branch-and-bound [Scozzari and Tardella, 2008], [Liuzzi et al.,2017]; KKT-based branch-and-bound [Burer and Vandenbussche, 2009], [Chen and Burer, 2012]

Motivation II

- General purpose solvers (e.g. BARON [Sahinidis, 1996] and COUENNE [Belotti, 2000]) usually exhibit poor performance.

Motivation II

- General purpose solvers (e.g. BARON [Sahinidis, 1996] and COUENNE [Belotti, 2000]) usually exhibit poor performance.
- We focus on MILP reformulations of standard quadratic programs.

Motivation II

- General purpose solvers (e.g. BARON [Sahinidis, 1996] and COUENNE [Belotti, 2000]) usually exhibit poor performance.
- We focus on MILP reformulations of standard quadratic programs.
- Sophisticated state-of-the-art MILP solvers (e.g., CPLEX, GUROBI, MOSEK, etc.)

Motivation II

- General purpose solvers (e.g. BARON [Sahinidis, 1996] and COUENNE [Belotti, 2000]) usually exhibit poor performance.
- We focus on MILP reformulations of standard quadratic programs.
- Sophisticated state-of-the-art MILP solvers (e.g., CPLEX, GUROBI, MOSEK, etc.)
- Our work is closely related to the MILP reformulation of nonconvex quadratic programs [Xia, Vera, and Zuluaga, 2015]

Motivation II

- General purpose solvers (e.g. BARON [Sahinidis, 1996] and COUENNE [Belotti, 2000]) usually exhibit poor performance.
- We focus on MILP reformulations of standard quadratic programs.
- Sophisticated state-of-the-art MILP solvers (e.g., CPLEX, GUROBI, MOSEK, etc.)
- Our work is closely related to the MILP reformulation of nonconvex quadratic programs [Xia, Vera, and Zuluaga, 2015]
- Our MILP reformulations are aimed at exploiting the specific structure of standard quadratic programs.

KKT Conditions

- Let $Q \in \mathcal{S}^n$.

$$(\text{StQP}) \quad \nu(Q) = \min\{x^T Qx : x \in \Delta_n\}.$$

KKT Conditions

- Let $Q \in \mathcal{S}^n$.
$$(\mathrm{StQP}) \quad \nu(Q) = \min\{x^T Qx : x \in \Delta_n\}.$$

- By the Karush-Kuhn-Tucker optimality conditions, if $x \in \Delta_n$ is an optimal solution of (StQP), then there exist $s \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ such that

$$\begin{aligned} Qx - \lambda e - s &= 0, && (1)\\ e^T x &= 1, && (2)\\ x &\in \mathbb{R}^n_+, && (3)\\ s &\in \mathbb{R}^n_+, && (4)\\ x_j s_j &= 0, \quad j = 1, \ldots, n. && (5) \end{aligned}$$

KKT Conditions

- Let $Q \in \mathcal{S}^n$.

$$(\mathrm{StQP}) \quad \nu(Q) = \min\{x^T Qx : x \in \Delta_n\}.$$

- By the Karush-Kuhn-Tucker optimality conditions, if $x \in \Delta_n$ is an optimal solution of (StQP), then there exist $s \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ such that

$$\begin{aligned} Qx - \lambda e - s &= 0, && (1)\\ e^T x &= 1, && (2)\\ x &\in \mathbb{R}^n_+, && (3)\\ s &\in \mathbb{R}^n_+, && (4)\\ x_j s_j &= 0, \quad j = 1, \ldots, n. && (5) \end{aligned}$$

- $x \in \Delta_n$ is a KKT point of (StQP) if there exists $(s, \lambda) \in \mathbb{R}^n \times \mathbb{R}$ such that (1) – (5) are satisfied.

KKT Conditions

- Let $Q \in \mathcal{S}^n$.

$$\text{(StQP)} \quad \nu(Q) = \min\{x^T Qx : x \in \Delta_n\}.$$

- By the Karush-Kuhn-Tucker optimality conditions, if $x \in \Delta_n$ is an optimal solution of (StQP), then there exist $s \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ such that

$$\begin{aligned} Qx - \lambda e - s &= 0, && (1)\\ e^T x &= 1, && (2)\\ x &\in \mathbb{R}^n_+, && (3)\\ s &\in \mathbb{R}^n_+, && (4)\\ x_j s_j &= 0, \quad j = 1, \ldots, n. && (5) \end{aligned}$$

- $x \in \Delta_n$ is a KKT point of (StQP) if there exists $(s, \lambda) \in \mathbb{R}^n \times \mathbb{R}$ such that (1) – (5) are satisfied.
- By (1), (2), and (5), if $x \in \Delta_n$ is a KKT point of (StQP), then $\nu(Q) = x^T Qx = \lambda$.

An LCP Reformulation

- Therefore, (StQP) can be equivalently reformulated by

$$
\begin{array}{llrcll}
\text{(R1)} & \min & \lambda & & & \\
 & & Qx - \lambda e - s & = & 0, & \\
 & & e^T x & = & 1, & \\
 & & x_j s_j & = & 0, & j = 1, \ldots, n, \\
 & & x & \geq & 0, & \\
 & & s & \geq & 0. &
\end{array}
$$

An LCP Reformulation

- Therefore, (StQP) can be equivalently reformulated by

$$
\begin{array}{llrcll}
\text{(R1)} & \min & \lambda & & & \\
& & Qx - \lambda e - s & = & 0, & \\
& & e^T x & = & 1, & \\
& & x_j s_j & = & 0, & j = 1, \ldots, n, \\
& & x & \geq & 0, & \\
& & s & \geq & 0. &
\end{array}
$$

- We can linearize the nonconvex complementarity constraints by using binary variables.

KKT-Based MILP Reformulation

$$
\begin{array}{llrcll}
\text{(MILP1)} & \min & \lambda & & & \\
& & Qx - \lambda e - s & = & 0, & \\
& & e^T x & = & 1, & \\
& & x_j & \leq & y_j, & j = 1, \ldots, n, \\
& & s_j & \leq & M_j(1 - y_j), & j = 1, \ldots, n, \\
& & x & \geq & 0, & \\
& & s & \geq & 0, & \\
& & y_j & \in & \{0, 1\}, & j = 1, \ldots, n.
\end{array}
$$

KKT-Based MILP Reformulation

$$
\begin{array}{llrcl}
\text{(MILP1)} & \min & \lambda & & \\
& & Qx - \lambda e - s & = & 0, \\
& & e^T x & = & 1, \\
& & x_j & \leq & y_j, \quad j = 1, \ldots, n, \\
& & s_j & \leq & M_j(1 - y_j), \quad j = 1, \ldots, n, \\
& & x & \geq & 0, \\
& & s & \geq & 0, \\
& & y_j & \in & \{0, 1\}, \quad j = 1, \ldots, n.
\end{array}
$$

- How big should M_j be?

Big M

- Recall

$$s_j \leq M_j(1 - y_j), \quad j = 1, \ldots, n.$$

Big M

- Recall

$$s_j \leq M_j(1 - y_j), \quad j = 1, \ldots, n.$$

- By the first constraint $Qx - \lambda e - s = 0$, which implies,

$$s_j = e_j^T Qx - \lambda, \quad j = 1, \ldots, n,$$

where $e_j \in \mathbb{R}^n$ denotes the jth unit vector, $j = 1, \ldots, n$.

Big M

- Recall

$$s_j \leq M_j(1 - y_j), \quad j = 1, \ldots, n.$$

- By the first constraint $Qx - \lambda e - s = 0$, which implies,

$$s_j = e_j^T Qx - \lambda, \quad j = 1, \ldots, n,$$

  where $e_j \in \mathbb{R}^n$ denotes the jth unit vector, $j = 1, \ldots, n$.
- Since $x \in \Delta_n$, we have $e_j^T Qx = x^T Qe_j \leq \max_{i=1,\ldots,n} Q_{ij}$.

Big M

- Recall

$$s_j \leq M_j(1 - y_j), \quad j = 1, \ldots, n.$$

- By the first constraint $Qx - \lambda e - s = 0$, which implies,

$$s_j = e_j^T Qx - \lambda, \quad j = 1, \ldots, n,$$

where $e_j \in \mathbb{R}^n$ denotes the jth unit vector, $j = 1, \ldots, n$.

- Since $x \in \Delta_n$, we have $e_j^T Qx = x^T Qe_j \leq \max_{i=1,\ldots,n} Q_{ij}$.
- If we can obtain a lower bound on λ (equivalently, a lower bound on $\nu(Q)$), then we can use it to obtain an upper bound on s_j, $j = 1, \ldots, n$.

A Simple Lower Bound on $\nu(Q)$

- Let $Q \in \mathcal{S}^n$ and let $\lambda \in \mathbb{R}$.

A Simple Lower Bound on $\nu(Q)$

- Let $Q \in \mathcal{S}^n$ and let $\lambda \in \mathbb{R}$.
- Suppose that $Q - \lambda ee^T$ satisfies $\nu(Q - \lambda ee^T) \geq 0$. Then,

A Simple Lower Bound on $\nu(Q)$

- Let $Q \in \mathcal{S}^n$ and let $\lambda \in \mathbb{R}$.
- Suppose that $Q - \lambda ee^T$ satisfies $\nu(Q - \lambda ee^T) \geq 0$. Then,

$$x^T \left(Q - \lambda ee^T\right) x = x^T Qx - \lambda \geq \nu(Q - \lambda ee^T) = \nu(Q) - \lambda \geq 0, \quad \forall x \in \Delta_n.$$

A Simple Lower Bound on $\nu(Q)$

- Let $Q \in \mathcal{S}^n$ and let $\lambda \in \mathbb{R}$.
- Suppose that $Q - \lambda ee^T$ satisfies $\nu(Q - \lambda ee^T) \geq 0$. Then,

$$x^T \left(Q - \lambda ee^T \right) x = x^T Qx - \lambda \geq \nu(Q - \lambda ee^T) = \nu(Q) - \lambda \geq 0, \quad \forall x \in \Delta_n.$$

- Therefore, if $\nu(Q - \lambda ee^T) \geq 0$, then $\nu(Q) \geq \lambda$.

A Simple Lower Bound on $\nu(Q)$

- Let $Q \in \mathcal{S}^n$ and let $\lambda \in \mathbb{R}$.
- Suppose that $Q - \lambda ee^T$ satisfies $\nu(Q - \lambda ee^T) \geq 0$. Then,

$$x^T\left(Q - \lambda ee^T\right)x = x^TQx - \lambda \geq \nu(Q - \lambda ee^T) = \nu(Q) - \lambda \geq 0, \quad \forall x \in \Delta_n.$$

- Therefore, if $\nu(Q - \lambda ee^T) \geq 0$, then $\nu(Q) \geq \lambda$.
- How can we ensure that $\nu(Q - \lambda ee^T) \geq 0$?

A Simple Lower Bound on $\nu(Q)$

- Let $Q \in \mathcal{S}^n$ and let $\lambda \in \mathbb{R}$.
- Suppose that $Q - \lambda ee^T$ satisfies $\nu(Q - \lambda ee^T) \geq 0$. Then,

$$x^T \left(Q - \lambda ee^T \right) x = x^T Qx - \lambda \geq \nu(Q - \lambda ee^T) = \nu(Q) - \lambda \geq 0, \quad \forall x \in \Delta_n.$$

- Therefore, if $\nu(Q - \lambda ee^T) \geq 0$, then $\nu(Q) \geq \lambda$.
- How can we ensure that $\nu(Q - \lambda ee^T) \geq 0$?
- If $Q - \lambda ee^T$ has nonnegative components, then $\nu(Q - \lambda ee^T) \geq 0$, since

$$x^T \left(Q - \lambda ee^T \right) x = x^T Qx - \lambda \geq 0, \quad \forall x \in \Delta_n.$$

A Simple Lower Bound on $\nu(Q)$

- The best lower bound is given by

$$\nu(Q) \quad \geq \quad \mathsf{sup}\{\lambda : Q - \lambda ee^T \text{ has nonnegative components}\},$$

A Simple Lower Bound on $\nu(Q)$

- The best lower bound is given by

$$\begin{aligned} \nu(Q) &\geq \sup\{\lambda : Q - \lambda e e^T \text{ has nonnegative components}\}, \\ &= \sup\{\lambda : Q_{ij} \geq \lambda, \quad i = 1, \ldots, n;\ j = 1, \ldots, n\}. \end{aligned}$$

A Simple Lower Bound on $\nu(Q)$

- The best lower bound is given by

$$\begin{aligned} \nu(Q) &\geq \sup\{\lambda : Q - \lambda e e^T \text{ has nonnegative components}\}, \\ &= \sup\{\lambda : Q_{ij} \geq \lambda, \quad i = 1, \ldots, n; \ j = 1, \ldots, n\}. \end{aligned}$$

- Clearly, the best lower bound is given by $\lambda_1 = \min_{1 \leq i \leq j \leq n} Q_{ij}$.

A Simple Lower Bound on $\nu(Q)$

- The best lower bound is given by

$$\begin{aligned} \nu(Q) &\geq \sup\{\lambda : Q - \lambda ee^T \text{ has nonnegative components}\}, \\ &= \sup\{\lambda : Q_{ij} \geq \lambda, \quad i = 1, \ldots, n; \ j = 1, \ldots, n\}. \end{aligned}$$

- Clearly, the best lower bound is given by $\lambda_1 = \min_{1 \leq i \leq j \leq n} Q_{ij}$.
- We obtain $\nu(Q) \geq \lambda_1 = \min_{1 \leq i \leq j \leq n} Q_{ij}$.

A Simple Lower Bound on $\nu(Q)$

- The best lower bound is given by

$$\begin{aligned} \nu(Q) &\geq \sup\{\lambda : Q - \lambda ee^T \text{ has nonnegative components}\}, \\ &= \sup\{\lambda : Q_{ij} \geq \lambda, \quad i = 1, \ldots, n;\ j = 1, \ldots, n\}. \end{aligned}$$

- Clearly, the best lower bound is given by $\lambda_1 = \min_{1 \leq i \leq j \leq n} Q_{ij}$.
- We obtain $\nu(Q) \geq \lambda_1 = \min_{1 \leq i \leq j \leq n} Q_{ij}$.
- This lower bound can be slightly sharpened if $Q - \lambda_1 ee^T$ has strictly positive entries along the main diagonal. [Bomze et al., 2008].

A Simple Lower Bound on $\nu(Q)$

- The best lower bound is given by

$$\begin{aligned}\nu(Q) &\geq \sup\{\lambda : Q - \lambda ee^T \text{ has nonnegative components}\},\\ &= \sup\{\lambda : Q_{ij} \geq \lambda, \quad i = 1, \ldots, n;\ j = 1, \ldots, n\}.\end{aligned}$$

- Clearly, the best lower bound is given by $\lambda_1 = \min_{1 \leq i \leq j \leq n} Q_{ij}$.
- We obtain $\nu(Q) \geq \lambda_1 = \min_{1 \leq i \leq j \leq n} Q_{ij}$.
- This lower bound can be slightly sharpened if $Q - \lambda_1 ee^T$ has strictly positive entries along the main diagonal. [Bomze et al., 2008].
- Henceforth, $\ell_1(Q)$ denotes the slightly sharpened lower bound, i.e.,

$$\text{(LB1)} \quad \nu(Q) \geq \ell_1(Q) := \min_{1 \leq i \leq j \leq n} Q_{ij} + \frac{1}{\sum_{k=1}^{n} (1/(Q_{kk} - \min_{1 \leq i \leq j \leq n} Q_{ij}))}.$$

A Tighter Lower Bound

- Let $Q \in \mathcal{S}^n$ and let $\lambda \in \mathbb{R}$.

A Tighter Lower Bound

- Let $Q \in \mathcal{S}^n$ and let $\lambda \in \mathbb{R}$.
- Recall: If $\nu(Q - \lambda ee^T) \geq 0$, then $\nu(Q) \geq \lambda$.

A Tighter Lower Bound

- Let $Q \in \mathcal{S}^n$ and let $\lambda \in \mathbb{R}$.
- Recall: If $\nu(Q - \lambda ee^T) \geq 0$, then $\nu(Q) \geq \lambda$.
- Can we obtain a larger class of matrices that satisfy $\nu(Q - \lambda ee^T) \geq 0$?

A Tighter Lower Bound

- Let $Q \in \mathcal{S}^n$ and let $\lambda \in \mathbb{R}$.
- Recall: If $\nu(Q - \lambda ee^T) \geq 0$, then $\nu(Q) \geq \lambda$.
- Can we obtain a larger class of matrices that satisfy $\nu(Q - \lambda ee^T) \geq 0$?
- If $\nu(Q - \lambda ee^T) = S_1 + S_2$, where $S_1 \in \mathcal{S}^n$ is positive semidefinite and $S_2 \in \mathcal{S}^n$ has nonnegative components, then $\nu(Q - \lambda ee^T) \geq 0$ (such matrices are called SPN).

A Tighter Lower Bound

- Let $Q \in \mathcal{S}^n$ and let $\lambda \in \mathbb{R}$.
- Recall: If $\nu(Q - \lambda ee^T) \geq 0$, then $\nu(Q) \geq \lambda$.
- Can we obtain a larger class of matrices that satisfy $\nu(Q - \lambda ee^T) \geq 0$?
- If $\nu(Q - \lambda ee^T) = S_1 + S_2$, where $S_1 \in \mathcal{S}^n$ is positive semidefinite and $S_2 \in \mathcal{S}^n$ has nonnegative components, then $\nu(Q - \lambda ee^T) \geq 0$ (such matrices are called SPN).
- The best lower bound is given by

$$(\mathrm{LB2}) \quad \nu(Q) \geq \ell_2(Q) := \max\{\lambda : Q - \lambda ee^T \text{ is SPN}\}.$$

Comparison of Lower Bounds

- Recall

$$\nu(Q) \geq \ell_1(Q) = \min_{1 \leq i \leq j \leq n} Q_{ij} + \frac{1}{\sum_{k=1}^{n} (1/(Q_{kk} - \min_{1 \leq i \leq j \leq n} Q_{ij}))},$$

$$\nu(Q) \geq \ell_2(Q) = \max\{\lambda : Q - \lambda e e^T \text{ is SPN}\}.$$

Comparison of Lower Bounds

- Recall

$$\nu(Q) \geq \ell_1(Q) = \min_{1\leq i\leq j\leq n} Q_{ij} + \frac{1}{\sum_{k=1}^{n}(1/(Q_{kk} - \min_{1\leq i\leq j\leq n} Q_{ij}))},$$
$$\nu(Q) \geq \ell_2(Q) = \max\{\lambda : Q - \lambda ee^T \text{ is SPN}\}.$$

- We have $\ell_1(Q) \leq \ell_2(Q)$.

Comparison of Lower Bounds

- Recall

$$\nu(Q) \geq \ell_1(Q) = \min_{1 \leq i \leq j \leq n} Q_{ij} + \frac{1}{\sum_{k=1}^{n} (1/(Q_{kk} - \min_{1 \leq i \leq j \leq n} Q_{ij}))},$$

$$\nu(Q) \geq \ell_2(Q) = \max\{\lambda : Q - \lambda e e^T \text{ is SPN}\}.$$

- We have $\ell_1(Q) \leq \ell_2(Q)$.
- $\ell_1(Q)$ can be computed in $O(n^2)$ time whereas $\ell_2(Q)$ can be computed by solving a semidefinite program.

Comparison of Lower Bounds

- Recall

$$\begin{aligned} \nu(Q) &\geq \ell_1(Q) = \min_{1 \leq i \leq j \leq n} Q_{ij} + \frac{1}{\sum_{k=1}^{n} (1/(Q_{kk} - \min_{1 \leq i \leq j \leq n} Q_{ij}))}, \\ \nu(Q) &\geq \ell_2(Q) = \max\{\lambda : Q - \lambda e e^T \text{ is SPN}\}. \end{aligned}$$

- We have $\ell_1(Q) \leq \ell_2(Q)$.
- $\ell_1(Q)$ can be computed in $O(n^2)$ time whereas $\ell_2(Q)$ can be computed by solving a semidefinite program.
- There exist other lower bounds in the literature (see, e.g., [Nowak, 1999], [Bomze and de Klerk, 2002], [Anstreicher and Burer, 2005]; and [Bomze et al., 2008] for a comparison)

Comparison of Lower Bounds

- Recall

$$\begin{aligned}\nu(Q) &\geq \ell_1(Q) = \min_{1\leq i\leq j\leq n} Q_{ij} + \frac{1}{\sum_{k=1}^{n}(1/(Q_{kk} - \min_{1\leq i\leq j\leq n} Q_{ij}))},\\ \nu(Q) &\geq \ell_2(Q) = \max\{\lambda : Q - \lambda ee^T \text{ is SPN}\}.\end{aligned}$$

- We have $\ell_1(Q) \leq \ell_2(Q)$.
- $\ell_1(Q)$ can be computed in $O(n^2)$ time whereas $\ell_2(Q)$ can be computed by solving a semidefinite program.
- There exist other lower bounds in the literature (see, e.g., [Nowak, 1999], [Bomze and de Klerk, 2002], [Anstreicher and Burer, 2005]; and [Bomze et al., 2008] for a comparison)
- Usually, trade-off between quality and computational cost.

Back to KKT-Based MILP Formulation

- Recall the KKT-based MILP formulation of (StQP):

$$
\begin{array}{llrcl}
\text{(MILP1)} & \min & \lambda & & \\
& & Qx - \lambda e - s & = & 0, \\
& & e^T x & = & 1, \\
& & x_j & \leq & y_j, \quad j = 1, \ldots, n, \\
& & s_j & \leq & M_j(1 - y_j), \quad j = 1, \ldots, n, \\
& & x & \geq & 0, \\
& & s & \geq & 0, \\
& & y_j & \in & \{0, 1\}, \quad j = 1, \ldots, n.
\end{array}
$$

Back to KKT-Based MILP Formulation

- Recall the KKT-based MILP formulation of (StQP):

$$
\begin{array}{llrcll}
\text{(MILP1)} & \min & \lambda & & & \\
& & Qx - \lambda e - s & = & 0, & \\
& & e^T x & = & 1, & \\
& & x_j & \leq & y_j, & j = 1, \ldots, n, \\
& & s_j & \leq & M_j(1 - y_j), & j = 1, \ldots, n, \\
& & x & \geq & 0, & \\
& & s & \geq & 0, & \\
& & y_j & \in & \{0, 1\}, & j = 1, \ldots, n.
\end{array}
$$

- For any feasible solution (x, y, s, λ), we have

$$
\begin{array}{rcll}
\lambda & \geq & \nu(Q), & \\
s_j & = & e_j^T Qx - \lambda, & j = 1, \ldots, n.
\end{array}
$$

Choice of Big M

- Therefore,

$$s_j = x^T Q e_j - \lambda \leq \max_{i=1,\ldots,n} Q_{ij} - \nu(Q) \leq \max_{i=1,\ldots,n} Q_{ij} - \ell,$$

where ℓ is any lower bound on $\nu(Q)$.

Choice of Big M

- Therefore,

$$s_j = x^T Q e_j - \lambda \le \max_{i=1,\ldots,n} Q_{ij} - \nu(Q) \le \max_{i=1,\ldots,n} Q_{ij} - \ell,$$

  where ℓ is any lower bound on $\nu(Q)$.
- Therefore, we can substitute $M_j := \max_{i=1,\ldots,n} Q_{ij} - \ell$ in the constraint $s_j \le M_j(1 - y_j), \ j = 1, \ldots, n$.

Choice of Big M

- Therefore,

$$s_j = x^T Q e_j - \lambda \leq \max_{i=1,\ldots,n} Q_{ij} - \nu(Q) \leq \max_{i=1,\ldots,n} Q_{ij} - \ell,$$

  where ℓ is any lower bound on $\nu(Q)$.
- Therefore, we can substitute $M_j := \max_{i=1,\ldots,n} Q_{ij} - \ell$ in the constraint $s_j \leq M_j(1 - y_j), \; j = 1, \ldots, n$.
- In theory, one can use any lower bound ℓ on $\nu(Q)$.

Choice of Big M

- Therefore,

$$s_j = x^T Q e_j - \lambda \leq \max_{i=1,\ldots,n} Q_{ij} - \nu(Q) \leq \max_{i=1,\ldots,n} Q_{ij} - \ell,$$

  where ℓ is any lower bound on $\nu(Q)$.
- Therefore, we can substitute $M_j := \max_{i=1,\ldots,n} Q_{ij} - \ell$ in the constraint $s_j \leq M_j(1 - y_j), \; j = 1, \ldots, n$.
- In theory, one can use any lower bound ℓ on $\nu(Q)$.
- In practice, however, larger values of big M tend to yield poorer linear programming relaxations and may lead to numerical instability.

A Different Perspective

- For $x \in \Delta_n$, the support set is given by

$$\mathcal{P}(x) = \{j \in \{1, \ldots, n\} : x_j > 0\}.$$

A Different Perspective

- For $x \in \Delta_n$, the support set is given by

$$\mathcal{P}(x) = \{j \in \{1, \ldots, n\} : x_j > 0\}.$$

Lemma

Let $x \in \Delta_n$ and let $Q \in \mathcal{S}^n$. Then,

$$\min_{j \in \mathcal{P}(x)} e_j^T Qx \leq x^T Qx \leq \max_{j \in \mathcal{P}(x)} e_j^T Qx.$$

A Different Perspective

- For $x \in \Delta_n$, the support set is given by

$$\mathcal{P}(x) = \{j \in \{1, \ldots, n\} : x_j > 0\}.$$

Lemma

Let $x \in \Delta_n$ and let $Q \in \mathcal{S}^n$. Then,

$$\min_{j \in \mathcal{P}(x)} e_j^T Qx \leq x^T Qx \leq \max_{j \in \mathcal{P}(x)} e_j^T Qx.$$

Furthermore, if $x \in \Delta_n$ is a KKT point of (StQP), then

$$\min_{j \in \mathcal{P}(x)} e_j^T Qx = x^T Qx = \max_{j \in \mathcal{P}(x)} e_j^T Qx.$$

A Min-Max Characterization

Proposition

Given an instance of (StQP),

$$\nu(Q) = \min\{x^T Qx : x \in \Delta_n\} = \min_{x \in \Delta_n} \max_{j \in \mathcal{P}(x)} e_j^T Qx.$$

An Alternative Min-Max MILP Formulation

$$
\begin{array}{llrcl}
\text{(MILP2)} & \min & \alpha & & \\
& & e_j^T Qx & \leq & \alpha + z_j, \quad j = 1, \ldots, n, \\
& & e^T x & = & 1, \\
& & x_j & \leq & y_j, \quad j = 1, \ldots, n, \\
& & z_j & \leq & U_j(1 - y_j), \quad j = 1, \ldots, n, \\
& & x & \geq & 0, \\
& & z & \geq & 0, \\
& & y_j & \in & \{0, 1\}, \quad j = 1, \ldots, n.
\end{array}
$$

An Alternative Min-Max MILP Formulation

$$
\begin{array}{llrcl}
\text{(MILP2)} & \min & \alpha & & \\
& & e_j^T Qx & \leq & \alpha + z_j, \quad j = 1, \ldots, n, \\
& & e^T x & = & 1, \\
& & x_j & \leq & y_j, \quad j = 1, \ldots, n, \\
& & z_j & \leq & U_j(1 - y_j), \quad j = 1, \ldots, n, \\
& & x & \geq & 0, \\
& & z & \geq & 0, \\
& & y_j & \in & \{0, 1\}, \quad j = 1, \ldots, n.
\end{array}
$$

Remark

Given an instance of (StQP), (MILP2) is an equivalent reformulation of (StQP) if

$$U_j \geq M_j, \quad j = 1, \ldots, n,$$

where $M_j = \max_{i=1,\ldots,n} Q_{ij} - \ell$ *and* ℓ *is any valid lower bound on* $\nu(Q)$.

A Comparison of Two Formulations

- There is a one-to-one correspondence between feasible solutions (x, y, s, λ) of the KKT-based formulation (MILP1) and KKT points of (StQP).

A Comparison of Two Formulations

- There is a one-to-one correspondence between feasible solutions (x, y, s, λ) of the KKT-based formulation (MILP1) and KKT points of (StQP).
- On the other hand, for any $x \in \Delta_n$, we can construct a feasible solution (x, y, z, α) of the min-max based formulation (MILP2) such that $\alpha \geq x^T Qx$, with equality if x is a KKT point of (StQP).

A Comparison of Two Formulations

- There is a one-to-one correspondence between feasible solutions (x, y, s, λ) of the KKT-based formulation (MILP1) and KKT points of (StQP).
- On the other hand, for any $x \in \Delta_n$, we can construct a feasible solution (x, y, z, α) of the min-max based formulation (MILP2) such that $\alpha \geq x^T Qx$, with equality if x is a KKT point of (StQP).
- Therefore, (MILP2) is an *exact relaxation* of (MILP1).

Convexity Graph

$$(\mathrm{StQP}) \quad \nu(Q) = \min\{x^T Qx : x \in \Delta_n\}$$

Convexity Graph

$$(\mathrm{StQP}) \quad \nu(Q) = \min\{x^T Qx : x \in \Delta_n\}$$

- Δ_n has n vertices $e_1, \ldots, e_n$ and $\binom{n}{2}$ edges.

Convexity Graph

$$\text{(StQP)} \quad \nu(Q) = \min\{x^T Qx : x \in \Delta_n\}$$

- Δ_n has n vertices $e_1, \ldots, e_n$ and $\binom{n}{2}$ edges.
- The restriction of $x^T Qx$ along the edge between e_i and e_j is strictly convex iff $Q_{ii} + Q_{jj} - 2Q_{ij} > 0, \; 1 \leq i < j \leq n$.

Convexity Graph

$$(\mathrm{StQP}) \quad \nu(Q) = \min\{x^T Qx : x \in \Delta_n\}$$

- Δ_n has n vertices $e_1, \ldots, e_n$ and $\binom{n}{2}$ edges.
- The restriction of $x^T Qx$ along the edge between e_i and e_j is strictly convex iff $Q_{ii} + Q_{jj} - 2Q_{ij} > 0,\ 1 \leq i < j \leq n$.

Definition

A graph $G = (V, E)$ is called the *convexity graph* of Q if $V = \{1, \ldots, n\}$ and

$$E = \{(i, j) : Q_{ii} + Q_{jj} - 2Q_{ij} > 0, \quad 1 \leq i < j \leq n\}.$$

Convexity Graph

$$(\mathrm{StQP}) \quad \nu(Q) = \min\{x^T Qx : x \in \Delta_n\}$$

- Δ_n has n vertices $e_1, \ldots, e_n$ and $\binom{n}{2}$ edges.
- The restriction of $x^T Qx$ along the edge between e_i and e_j is strictly convex iff $Q_{ii} + Q_{jj} - 2Q_{ij} > 0, \; 1 \leq i < j \leq n$.

Definition

A graph $G = (V, E)$ is called the *convexity graph* of Q if $V = \{1, \ldots, n\}$ and

$$E = \{(i,j) : Q_{ii} + Q_{jj} - 2Q_{ij} > 0, \quad 1 \leq i < j \leq n\}.$$

Theorem (Scozzari and Tardella, 2008)

There exists an optimal solution $x^* \in \Delta_n$ of (StQP) such that the vertices corrresponding to $\mathcal{P}(x^*)$ form a clique in the convexity graph $G = (V, E)$ of Q.

Valid Inequalities

- Recall the convexity graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ and

$$E = \{(i,j) : Q_{ii} + Q_{jj} - 2Q_{ij} > 0, \quad 1 \leq i < j \leq n\}.$$

Valid Inequalities

- Recall the convexity graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ and

$$E = \{(i,j) : Q_{ii} + Q_{jj} - 2Q_{ij} > 0, \quad 1 \le i < j \le n\}.$$

- There is an optimal solution $x^* \in \Delta_n$ of (StQP) such that the vertices corresponding to $\mathcal{P}(x^*)$ form a clique in $G = (V, E)$.

Valid Inequalities

- Recall the convexity graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ and

$$E = \{(i, j) : Q_{ii} + Q_{jj} - 2Q_{ij} > 0, \quad 1 \leq i < j \leq n\}.$$

- There is an optimal solution $x^* \in \Delta_n$ of (StQP) such that the vertices corresponding to $\mathcal{P}(x^*)$ form a clique in $G = (V, E)$.
- Vertices corresponding to $\mathcal{P}(x^*)$ form a stable (independent) set in the complement of G.

Valid Inequalities

- Recall the convexity graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ and

$$E = \{(i,j) : Q_{ii} + Q_{jj} - 2Q_{ij} > 0, \quad 1 \leq i < j \leq n\}.$$

- There is an optimal solution $x^* \in \Delta_n$ of (StQP) such that the vertices corresponding to $\mathcal{P}(x^*)$ form a clique in $G = (V, E)$.
- Vertices corresponding to $\mathcal{P}(x^*)$ form a stable (independent) set in the complement of G.

Theorem

The following inequalities are valid for both (MILP1) and (MILP2):

$$y_i + y_j \leq 1, \quad 1 \leq i < j \leq n \ s.t. \ Q_{ii} + Q_{jj} - 2Q_{ij} \leq 0.$$

Computational Experiment I

- No standard test problems for (StQP)

Computational Experiment I

- No standard test problems for (StQP)
- We used (StQP) instances generated by [Nowak, 1999] (later used by [Scozzari and Tardella, 2008] and [Liuzzi et al., 2017]; made publicly available by Giampaolo Liuzzi)

Computational Experiment I

- No standard test problems for (StQP)
- We used (StQP) instances generated by [Nowak, 1999] (later used by [Scozzari and Tardella, 2008] and [Liuzzi et al., 2017]; made publicly available by Giampaolo Liuzzi)
- Randomly generated instances with convexity graphs having a prespecified density δ ($n \in \{100, 200\}$ and $\delta \in \{0.25, 0.5, 0.75\}$).

Computational Experiment I

- No standard test problems for (StQP)
- We used (StQP) instances generated by [Nowak, 1999] (later used by [Scozzari and Tardella, 2008] and [Liuzzi et al., 2017]; made publicly available by Giampaolo Liuzzi)
- Randomly generated instances with convexity graphs having a prespecified density δ ($n \in \{100, 200\}$ and $\delta \in \{0.25, 0.5, 0.75\}$).
- Difficulty increases as δ increases.

Computational Experiment I

- No standard test problems for (StQP)
- We used (StQP) instances generated by [Nowak, 1999] (later used by [Scozzari and Tardella, 2008] and [Liuzzi et al., 2017]; made publicly available by Giampaolo Liuzzi)
- Randomly generated instances with convexity graphs having a prespecified density δ ($n \in \{100, 200\}$ and $\delta \in \{0.25, 0.5, 0.75\}$).
- Difficulty increases as δ increases.
- Six instances for each choice of (n, δ) (total of 36 instances)

Computational Setup

- Two MILP formulations (KKT-Based and Min-Max Based)

Computational Setup

- Two MILP formulations (KKT-Based and Min-Max Based)
- Two lower bounds ($\ell_1(Q)$ and $\ell_2(Q)$)

Computational Setup

- Two MILP formulations (KKT-Based and Min-Max Based)
- Two lower bounds ($\ell_1(Q)$ and $\ell_2(Q)$)
- All valid inequalities vs no valid inequalities

Computational Setup

- Two MILP formulations (KKT-Based and Min-Max Based)
- Two lower bounds ($\ell_1(Q)$ and $\ell_2(Q)$)
- All valid inequalities vs no valid inequalities
- Our implementation uses the YALMIP interface in (MATLAB 2017b). We use CPLEX 12.8 as an MILP solver and MOSEK 8) as an SDP solver.

Computational Setup

- Two MILP formulations (KKT-Based and Min-Max Based)
- Two lower bounds ($\ell_1(Q)$ and $\ell_2(Q)$)
- All valid inequalities vs no valid inequalities
- Our implementation uses the YALMIP interface in (MATLAB 2017b). We use CPLEX 12.8 as an MILP solver and MOSEK 8) as an SDP solver.
- Four Intel Xeon CPUs (E5-4610 @ 1.80 Ghz), 10 cores per socket, 2 threads per core, 512 GB RAM, Redhat Enterprise Linux.

Computational Setup

- Two MILP formulations (KKT-Based and Min-Max Based)
- Two lower bounds ($\ell_1(Q)$ and $\ell_2(Q)$)
- All valid inequalities vs no valid inequalities
- Our implementation uses the YALMIP interface in (MATLAB 2017b). We use CPLEX 12.8 as an MILP solver and MOSEK 8) as an SDP solver.
- Four Intel Xeon CPUs (E5-4610 @ 1.80 Ghz), 10 cores per socket, 2 threads per core, 512 GB RAM, Redhat Enterprise Linux.
- We imposed a CPU time limit of 3600 seconds for MILP problems (CPLEX uses at most 32 threads).

Computational Setup

- Two MILP formulations (KKT-Based and Min-Max Based)
- Two lower bounds ($\ell_1(Q)$ and $\ell_2(Q)$)
- All valid inequalities vs no valid inequalities
- Our implementation uses the YALMIP interface in (MATLAB 2017b). We use CPLEX 12.8 as an MILP solver and MOSEK 8) as an SDP solver.
- Four Intel Xeon CPUs (E5-4610 @ 1.80 Ghz), 10 cores per socket, 2 threads per core, 512 GB RAM, Redhat Enterprise Linux.
- We imposed a CPU time limit of 3600 seconds for MILP problems (CPLEX uses at most 32 threads).
- We used MATLAB's `fmincon` to compute a quick upper bound.

Computational Results I - Base Models

	KKT		MIN-MAX		DNN	`fmincon`
Instance	$\ell_1(Q)$	$\ell_2(Q)$	$\ell_1(Q)$	$\ell_2(Q)$	Time	Time
$(100, 0.25)$	0.62	0.49	0.38	**0.36**	14.04	1.72
$(100, 0.5)$	0.71	0.79	0.45	**0.44**	14.66	1.43
$(100, 0.75)$	2.28	2.03	0.86	**0.48**	15.49	1.10
$(200, 0.25)$	2.81	**1.46**	2.03	1.48	270.92	11.26
$(200, 0.5)$	14.65	11.83	7.74	**0.99**	297.44	7.98
$(200, 0.75)$	67.70	88.55	14.56	**2.07**	350.84	5.05

Table: Average Results

Computational Results I - Valid Inequalities

		KKT		MIN-MAX		DNN	`fmincon`
Instance	# of VIs	$\ell_1(Q)$	$\ell_2(Q)$	$\ell_1(Q)$	$\ell_2(Q)$	Time	Time
$(100, 0.25)$	3712	1.33	0.96	0.97	**0.75**	14.04	1.72
$(100, 0.5)$	2484	1.09	1.13	**0.70**	0.79	14.66	1.43
$(100, 0.75)$	1219	2.52	2.24	1.03	**0.73**	15.49	1.10
$(200, 0.25)$	14923	9.83	7.91	8.06	**4.86**	270.92	11.26
$(200, 0.5)$	9958	27.75	9.85	11.83	**4.28**	297.44	7.98
$(200, 0.75)$	4914	76.19	83.83	31.46	**6.99**	350.84	5.05

Table: Average Results

Overall Comparison and Discussion

	KKT		MIN-MAX	
	$\ell_1(Q)$	$\ell_2(Q)$	$\ell_1(Q)$	$\ell_2(Q)$
No VIs	532.57	630.82	156.13	**34.99**
VIs	712.30	635.51	324.19	110.38

Table: Cumulative Results

- Total time for solving the DNN relaxation is 5780.35.

Overall Comparison and Discussion

	KKT		MIN-MAX	
	$\ell_1(Q)$	$\ell_2(Q)$	$\ell_1(Q)$	$\ell_2(Q)$
No VIs	532.57	630.82	156.13	**34.99**
VIs	712.30	635.51	324.19	110.38

Table: Cumulative Results

- Total time for solving the DNN relaxation is 5780.35.
- Each matrix Q has about $n/2$ negative and $n/2$ positive eigenvalues.

Overall Comparison and Discussion

	KKT		MIN-MAX	
	$\ell_1(Q)$	$\ell_2(Q)$	$\ell_1(Q)$	$\ell_2(Q)$
No VIs	532.57	630.82	156.13	**34.99**
VIs	712.30	635.51	324.19	110.38

Table: Cumulative Results

- Total time for solving the DNN relaxation is 5780.35.
- Each matrix Q has about $n/2$ negative and $n/2$ positive eigenvalues.
- DNN relaxation was exact on all instances!

Overall Comparison and Discussion

	KKT		MIN-MAX	
	$\ell_1(Q)$	$\ell_2(Q)$	$\ell_1(Q)$	$\ell_2(Q)$
No VIs	532.57	630.82	156.13	**34.99**
VIs	712.30	635.51	324.19	110.38

Table: Cumulative Results

- Total time for solving the DNN relaxation is 5780.35.
- Each matrix Q has about $n/2$ negative and $n/2$ positive eigenvalues.
- DNN relaxation was exact on all instances!
- The support of optimal solutions was in the range $3, \ldots, 7$.

Computational Experiments

- We tested MILP formulations on the maximum stable set problem.

Computational Experiments

- We tested MILP formulations on the maximum stable set problem.
- Let $G = (V, E)$ be a simple, undirected graph.

Computational Experiments

- We tested MILP formulations on the maximum stable set problem.
- Let $G = (V, E)$ be a simple, undirected graph.
- A set $S \subseteq V$ is a **stable set** if each pair of vertices in S is mutually nonadjacent.

Computational Experiments

- We tested MILP formulations on the maximum stable set problem.
- Let $G = (V, E)$ be a simple, undirected graph.
- A set $S \subseteq V$ is a **stable set** if each pair of vertices in S is mutually nonadjacent.
- The maximum stable set problem is concerned with finding a stable set with the largest cardinality, denoted by $\alpha(G)$.

Computational Experiments

- We tested MILP formulations on the maximum stable set problem.
- Let $G = (V, E)$ be a simple, undirected graph.
- A set $S \subseteq V$ is a **stable set** if each pair of vertices in S is mutually nonadjacent.
- The maximum stable set problem is concerned with finding a stable set with the largest cardinality, denoted by $\alpha(G)$.
- [Motzkin and Straus, 1965]

$$\frac{1}{\alpha(G)} = \min \left\{ x^T (I + A_G) x : x \in \Delta_n \right\}$$

Computational Experiments

- We tested MILP formulations on the maximum stable set problem.
- Let $G = (V, E)$ be a simple, undirected graph.
- A set $S \subseteq V$ is a **stable set** if each pair of vertices in S is mutually nonadjacent.
- The maximum stable set problem is concerned with finding a stable set with the largest cardinality, denoted by $\alpha(G)$.
- [Motzkin and Straus, 1965]

$$\frac{1}{\alpha(G)} = \min\left\{x^T(I + A_G)x : x \in \Delta_n\right\}$$

- We also solve another ILP formulation:

$$\alpha(G) = \max\left\{\sum_{j=1}^{n} x_j : x_i + x_j \leq 1, \quad (i,j) \in E, \quad x_j \in \{0,1\}, \quad j = 1, \ldots, n\right\}$$

Computational Results I - Base Models

					KKT		MIN-MAX		ILP	DNN
Instance	n	$\|E\|$	δ	$\alpha(G)$	$\ell_1(Q)$	$\ell_2(Q)$	$\ell_1(Q)$	$\ell_2(Q)$	Time	Time
johnson8-2-4.co	28	168	0.56	4	0.28	0.22	0.19	0.15	**0.04**	0.15
MANN-a9.co	45	72	0.93	16	3.96	19.39	4.8	1.95	**0.16**	0.67
hamming6-4.co	64	1312	0.35	4	0.26	**0.13**	0.17	0.17	0.22	1.53
hamming6-2.co	64	192	0.9	32	2.13	0.46	0.25	0.21	**0.03**	1.88
johnson8-4-4.co	70	560	0.77	14	0.28	0.31	0.58	0.28	**0.03**	2.62
johnson16-2-4.co	120	1680	0.76	8	0.38	0.19	0.38	0.12	**0.04**	20.27
C125.9.co	125	787	0.9	34	(73%)	(9%)	(73%)	(9%)	**0.39**	44.74
keller4.co	171	5100	0.65	11	(21%)	222.59	26.28	32.58	**2.71**	179.32
c-fat200-1.co	200	18366	0.08	12	1.54	0.28	1.17	**0.22**	11.59	197.5
c-fat200-2.co	200	16665	0.16	24	1.59	**0.79**	1.39	1.11	9.57	228.37
c-fat-200-5.co	200	11427	0.43	58	1.56	1.5	**0.99**	1.08	4.34	228.15
brock200-2.co	200	10024	0.5	12	148.17	109.31	102.08	93.37	**13.41**	319.01
brock200-3.co	200	7852	0.61	15	3380.29	1473.09	929.61	1205.56	**34.16**	319.22
brock200-4.co	200	6811	0.66	17	(93%)	(24%)	(92%)	2125.48	**33.59**	323.92
brock200-1.co	200	5066	0.75	21	(91%)	(26%)	(90%)	(26%)	**62.47**	349.36
sanr200-0.7.co	200	6032	0.7	18	(92%)	(24%)	(92%)	(24%)	**57.42**	321.44
sanr200-0.9.co	200	2037	0.9	42	(79%)	(16%)	(80%)	(14%)	**51.26**	353.87
san200-0.7-2.co	200	5970	0.7	18	(93%)	(17%)	(93%)	(17%)	**1.07**	450.41
san200-0.7-1.co	200	5970	0.7	30	(92%)	(43%)	(85%)	(43%)	**0.33**	270.1

Table: DIMACS Instances

Computational Results I - Valid Inequalities

					KKT		MIN-MAX		ILP	DNN
Instance	n	$\|E\|$	δ	$\alpha(G)$	$\ell_1(Q)$	$\ell_2(Q)$	$\ell_1(Q)$	$\ell_2(Q)$	Time	Time
johnson8-2-4.co	28	168	0.56	4	0.16	**0.02**	0.21	0.21	0.04	0.15
MANN-a9.co	45	72	0.93	16	0.48	0.29	0.5	0.31	**0.16**	0.67
hamming6-4.co	64	1312	0.35	4	0.5	0.16	0.44	**0.2**	0.22	1.53
hamming6-2.co	64	192	0.9	32	0.39	0.5	0.24	0.24	**0.03**	1.88
johnson8-4-4.co	70	560	0.77	14	0.99	0.24	1.19	0.18	**0.03**	2.62
johnson16-2-4.co	120	1680	0.76	8	0.56	**0.03**	0.43	0.16	0.04	20.27
C125.9.co	125	787	0.9	34	5.21	9.52	5.32	2.15	**0.39**	44.74
keller4.co	171	5100	0.65	11	35.68	26.17	28.28	29.76	**2.71**	179.32
c-fat200-1.co	200	18366	0.08	12	14.25	1.85	43.98	**1.73**	11.59	197.5
c-fat200-2.co	200	16665	0.16	24	13.5	1.73	31.38	**1.65**	9.57	228.37
c-fat-200-5.co	200	11427	0.43	58	5.9	5.42	10.94	5.09	**4.34**	228.15
brock200-2.co	200	10024	0.5	12	64.86	75.64	77.25	60.27	**13.41**	319.01
brock200-3.co	200	7852	0.61	15	178.06	105.28	204.35	171.24	**34.16**	319.22
brock200-4.co	200	6811	0.66	17	197.58	193.29	245.55	197.95	**33.59**	323.92
brock200-1.co	200	5066	0.75	21	880.32	1690.36	528.01	499.21	**62.47**	349.36
sanr200-0.7.co	200	6032	0.7	18	308.89	231.84	380.09	227.66	**57.42**	321.44
sanr200-0.9.co	200	2037	0.9	42	(78%)	(14%)	2261.92	(14%)	**51.26**	353.87
san200-0.7-2.co	200	5970	0.7	18	34.19	9.32	25.1	9.45	**1.07**	450.41
san200-0.7-1.co	200	5970	0.7	30	25.06	62.11	58.43	143.72	**0.33**	270.1

Table: DIMACS Instances

Overall Comparison and Discussion

	KKT		MIN-MAX	
	$\ell_1(Q)$	$\ell_2(Q)$	$\ell_1(Q)$	$\ell_2(Q)$
No VIs	32496 (8)	27154 (7)	26427 (8)	25115 (7)
VIs	5383 (1)	6030 (1)	**3904** (0)	4965 (1)

Table: Cumulative Results on 19 DIMACS instances

- Total time for solving the DNN relaxation is 3613.

Overall Comparison and Discussion

	KKT		MIN-MAX	
	$\ell_1(Q)$	$\ell_2(Q)$	$\ell_1(Q)$	$\ell_2(Q)$
No VIs	32496 (8)	27154 (7)	26427 (8)	25115 (7)
VIs	5383 (1)	6030 (1)	**3904** (0)	4965 (1)

Table: Cumulative Results on 19 DIMACS instances

- Total time for solving the DNN relaxation is 3613.
- Total time for solving ILP is 283.

Concluding Remarks

- We considered solving standard quadratic programs by using two alternative MILP reformulations.

Concluding Remarks

- We considered solving standard quadratic programs by using two alternative MILP reformulations.
- We used two different upper bounds on the parameters $M_j, \ j = 1, \ldots, n$.

Concluding Remarks

- We considered solving standard quadratic programs by using two alternative MILP reformulations.
- We used two different upper bounds on the parameters $M_j, \; j = 1, \ldots, n$.
- We proposed valid inequalities.

Concluding Remarks

- We considered solving standard quadratic programs by using two alternative MILP reformulations.
- We used two different upper bounds on the parameters $M_j,\ j = 1, \ldots, n$.
- We proposed valid inequalities.
- Encouraging computational results on certain sets of instances.

Concluding Remarks

- We considered solving standard quadratic programs by using two alternative MILP reformulations.
- We used two different upper bounds on the parameters $M_j, \ j = 1, \dots, n$.
- We proposed valid inequalities.
- Encouraging computational results on certain sets of instances.
- Performance on hard instances ([Bomze et al., 2017]?

Concluding Remarks

- We considered solving standard quadratic programs by using two alternative MILP reformulations.
- We used two different upper bounds on the parameters $M_j, \; j = 1, \ldots, n$.
- We proposed valid inequalities.
- Encouraging computational results on certain sets of instances.
- Performance on hard instances ([Bomze et al., 2017]?
- Comparison with other branch-and-bound approaches?

Acknowledgements

- Jacek Gondzio
- Julian Hall