Global Solutions of Nonconvex Standard Quadratic Programs via Mixed Integer Linear Programming Formulations

E. Alper Yıldırım

Koç University
Department of Industrial Engineering
Istanbul, Turkey

Workshop on Optimization, Game Theory, and Data Analysis 2018
University of Vienna, Vienna, Austria

December 21, 2018

Joint work with Jacek Gondzio

Outline

1. Introduction
   - Standard Quadratic Programs
2. Two MILP Formulations
   - KKT-Based Reformulation
   - Upper Bounds on Big-M Parameters
   - An Alternative MILP Formulation
   - Valid Inequalities
3. Computational Results
4. Conclusions

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

- $\Delta_n = \{x \in \mathbb{R}^n : e^T x = 1, \quad x \in \mathbb{R}^n_+\}$ (the unit simplex),
- $Q \in \mathcal{S}^n$, 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 $\mathbb{R}^n$.

Applications

- Portfolio optimization [Markowitz, 1952]
- Quadratic resource allocation problem [Ibaraki and Katoh, 1988]
- Population genetics [Kingman, 1961]
- Evolutionary game theory [Bomze, 2002]
- Social network analysis [Bomze et al., 2018]
- Copositivity detection (a matrix $M \in \mathcal{S}^n$ is copositive iff $\nu(M) = \min\{x^T Mx : x \in \Delta_n\} \geq 0$)
- Maximum (weighted) stable set problem [Motzkin and Straus, 1965], [Gibbons et al., 1997]

Applications

- Portfolio optimization [Markowitz, 1952]
- Quadratic resource allocation problem [Ibaraki and Katoh, 1988]
- Population genetics [Kingman, 1961]
- Evolutionary game theory [Bomze, 2002]
- Social network analysis [Bomze et al., 2018]
- Copositivity detection (a matrix $M \in \mathcal{S}^n$ is copositive iff $\nu(M) = \min\{x^T Mx : x \in \Delta_n\} \geq 0$)
- Maximum (weighted) stable set problem [Motzkin and Straus, 1965], [Gibbons et al., 1997]
- NP-hard in general

Focus and Previous Work

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

Focus and Previous Work

- In this talk, we are interested in solving (StQP) to global optimality.
- DC-based branch-and-bound [Bomze, 2002]; clique-based branch-and-bound [Scozzari and Tardella, 2008], [Liuzzi et al.,2017]; spatial branch-and-bound with cutting planes [Bonami et al., 2016]; KKT-based branch-and-bound [Burer and Vandenbussche, 2009], [Chen and Burer, 2012]

Focus and Previous Work

- In this talk, we are interested in solving (StQP) to global optimality.
- DC-based branch-and-bound [Bomze, 2002]; clique-based branch-and-bound [Scozzari and Tardella, 2008], [Liuzzi et al.,2017]; spatial branch-and-bound with cutting planes [Bonami et al., 2016]; KKT-based branch-and-bound [Burer and Vandenbussche, 2009], [Chen and Burer, 2012]
- LPCC-based approaches (linear programming with complementarity constraints) [Hu et al., 2012]

Focus and Previous Work

- In this talk, we are interested in solving (StQP) to global optimality.
- DC-based branch-and-bound [Bomze, 2002]; clique-based branch-and-bound [Scozzari and Tardella, 2008], [Liuzzi et al.,2017]; spatial branch-and-bound with cutting planes [Bonami et al., 2016]; KKT-based branch-and-bound [Burer and Vandenbussche, 2009], [Chen and Burer, 2012]
- LPCC-based approaches (linear programming with complementarity constraints) [Hu et al., 2012]
- MILP-based approaches [Hu et al., 2008], [Hu et al., 2012], [Yu et al., 2016], [Xia et al., 2015]

Focus and Previous Work

- In this talk, we are interested in solving (StQP) to global optimality.
- DC-based branch-and-bound [Bomze, 2002]; clique-based branch-and-bound [Scozzari and Tardella, 2008], [Liuzzi et al.,2017]; spatial branch-and-bound with cutting planes [Bonami et al., 2016]; KKT-based branch-and-bound [Burer and Vandenbussche, 2009], [Chen and Burer, 2012]
- LPCC-based approaches (linear programming with complementarity constraints) [Hu et al., 2012]
- MILP-based approaches [Hu et al., 2008], [Hu et al., 2012], [Yu et al., 2016], [Xia et al., 2015]
- Copositive reformulation – adaptive simplicial partitioning [Bundfuss and Dür, 2009]

Focus and Previous Work

- In this talk, we are interested in solving (StQP) to global optimality.
- DC-based branch-and-bound [Bomze, 2002]; clique-based branch-and-bound [Scozzari and Tardella, 2008], [Liuzzi et al.,2017]; spatial branch-and-bound with cutting planes [Bonami et al., 2016]; KKT-based branch-and-bound [Burer and Vandenbussche, 2009], [Chen and Burer, 2012]
- LPCC-based approaches (linear programming with complementarity constraints) [Hu et al., 2012]
- MILP-based approaches [Hu et al., 2008], [Hu et al., 2012], [Yu et al., 2016], [Xia et al., 2015]
- Copositive reformulation – adaptive simplicial partitioning [Bundfuss and Dür, 2009]
- General purpose solvers (e.g., BARON [Sahinidis, 1996], COUENNE [Belotti, 2000], CPLEX 12.6.0+)

Motivation

- Few specific global solution approaches for standard quadratic programs.

Motivation

- Few specific global solution approaches for standard quadratic programs.
- General purpose solvers usually exhibit poor performance.

Motivation

- Few specific global solution approaches for standard quadratic programs.
- General purpose solvers usually exhibit poor performance.
- Sophisticated state-of-the-art MILP solvers (e.g., CPLEX, GUROBI, MOSEK, etc.)

Motivation

- Few specific global solution approaches for standard quadratic programs.
- General purpose solvers usually exhibit poor performance.
- Sophisticated state-of-the-art MILP solvers (e.g., CPLEX, GUROBI, MOSEK, etc.)
- We focus on MILP reformulations of standard quadratic programs.

Motivation

- Few specific global solution approaches for standard quadratic programs.
- General purpose solvers usually exhibit poor performance.
- Sophisticated state-of-the-art MILP solvers (e.g., CPLEX, GUROBI, MOSEK, etc.)
- We focus on MILP reformulations of standard quadratic programs.
- Our work is related to the MILP-based approaches [Hu et al., 2008], [Hu et al., 2012], [Yu et al., 2016], [Xia et al., 2015]

Motivation

- Few specific global solution approaches for standard quadratic programs.
- General purpose solvers usually exhibit poor performance.
- Sophisticated state-of-the-art MILP solvers (e.g., CPLEX, GUROBI, MOSEK, etc.)
- We focus on MILP reformulations of standard quadratic programs.
- Our work is related to the MILP-based approaches [Hu et al., 2008], [Hu et al., 2012], [Yu et al., 2016], [Xia et al., 2015]
- Our MILP reformulations are aimed at exploiting the specific structure of standard quadratic programs.

Outline

KKT Conditions

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

KKT Conditions

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

- 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

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

KKT Conditions

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

- 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.

KKT Conditions

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

- 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 $\nu(Q) = x^T Qx = \lambda$.

KKT-Based MILP Reformulation

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

KKT-Based MILP Reformulation

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

- How big should K_j and M_j be?

Big-M Parameters

- Recall

$$\begin{aligned} x_j &\leq K_j y_j, & j &= 1, \ldots, n, \\ s_j &\leq M_j(1 - y_j), & j &= 1, \ldots, n. \end{aligned}$$

Big-M Parameters

- Recall

$$\begin{aligned} x_j &\leq K_j y_j, & j = 1, \ldots, n, \\ s_j &\leq M_j(1 - y_j), & j = 1, \ldots, n. \end{aligned}$$

- Since $e^T x = 1$ and $x \geq 0$, we have $x_j \leq 1, \; j = 1, \ldots, n$.

Big-M Parameters

- Recall

$$\begin{array}{rcll} x_j & \leq & K_j y_j, & j = 1, \ldots, n, \\ s_j & \leq & M_j(1 - y_j), & j = 1, \ldots, n. \end{array}$$

- Since $e^T x = 1$ and $x \geq 0$, we have $x_j \leq 1,\ j = 1, \ldots, n$.
- Choose $K_j = 1,\ j = 1, \ldots, n$.

Big-M Parameters

- Recall
$$\begin{aligned} x_j &\leq K_j y_j, & j = 1, \ldots, n, \\ s_j &\leq M_j(1 - y_j), & j = 1, \ldots, n. \end{aligned}$$
- Since $e^T x = 1$ and $x \geq 0$, we have $x_j \leq 1, \; j = 1, \ldots, n$.
- Choose $K_j = 1, \; j = 1, \ldots, n$.
- 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 Parameters

- Recall

$$\begin{aligned} x_j &\leq K_j y_j, & j = 1, \ldots, n, \\ s_j &\leq M_j(1 - y_j), & j = 1, \ldots, n. \end{aligned}$$

- Since $e^T x = 1$ and $x \geq 0$, we have $x_j \leq 1, \; j = 1, \ldots, n$.
- Choose $K_j = 1, \; j = 1, \ldots, n$.
- 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 Parameters

- Recall
$$\begin{array}{rcll} x_j & \leq & K_j y_j, & j = 1, \ldots, n, \\ s_j & \leq & M_j(1 - y_j), & j = 1, \ldots, n. \end{array}$$
- Since $e^T x = 1$ and $x \geq 0$, we have $x_j \leq 1,\ j = 1, \ldots, n$.
- Choose $K_j = 1,\ j = 1, \ldots, n$.
- 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 $e_j^T Qx = x^T Qe_j \leq \max_{i=1,\ldots,n} Q_{ij}$.
- Recall that $\lambda \geq \nu(Q)$ on the feasible region.

Big-M Parameters

- Recall

$$\begin{array}{rcll} x_j & \leq & K_j y_j, & j = 1, \ldots, n, \\ s_j & \leq & M_j(1 - y_j), & j = 1, \ldots, n. \end{array}$$

- Since $e^T x = 1$ and $x \geq 0$, we have $x_j \leq 1, \; j = 1, \ldots, n$.
- Choose $K_j = 1, \; j = 1, \ldots, n$.
- 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 $e_j^T Qx = x^T Qe_j \leq \max_{i=1,\ldots,n} Q_{ij}$.
- Recall that $\lambda \geq \nu(Q)$ on the feasible region.
- If we can obtain a lower bound on $\nu(Q)$, then we can use it to obtain an upper bound on $s_j, \; j = 1, \ldots, n$.

Two Lower Bounds

- Two lower bounds on $\nu(Q)$:

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

Two Lower Bounds

- Two lower bounds on $\nu(Q)$:

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

$$\nu(Q) \geq \ell_2(Q) = \min \left\{ \langle Q, X \rangle : \langle ee^T, X \rangle = 1, \quad X \in \mathcal{DN}^n \right\}.$$

Two Lower Bounds

- Two lower bounds on $\nu(Q)$:

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

$$\nu(Q) \geq \ell_2(Q) = \min \left\{ \langle Q, X \rangle : \langle ee^T, X \rangle = 1, \quad X \in \mathcal{DN}^n \right\}.$$

- We have $\ell_1(Q) \leq \ell_2(Q) \leq \nu(Q)$ [Bomze et al., 2008].

Two Lower Bounds

- Two lower bounds on $\nu(Q)$:

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

$$\nu(Q) \geq \ell_2(Q) = \min \left\{ \langle Q, X \rangle : \langle ee^T, X \rangle = 1, \quad X \in \mathcal{DN}^n \right\}.$$

- We have $\ell_1(Q) \leq \ell_2(Q) \leq \nu(Q)$ [Bomze et al., 2008].
- $\ell_1(Q)$ can be computed in $O(n^2)$ time whereas $\ell_2(Q)$ requires solving a computationally expensive semidefinite program.

Two Lower Bounds

- Two lower bounds on $\nu(Q)$:

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

$$\nu(Q) \geq \ell_2(Q) = \min \left\{ \langle Q, X\rangle : \langle ee^T, X\rangle = 1, \quad X \in \mathcal{DN}^n \right\}.$$

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

Two Lower Bounds

- Two lower bounds on $\nu(Q)$:

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

$$\nu(Q) \geq \ell_2(Q) = \min\left\{\langle Q, X\rangle : \langle ee^T, X\rangle = 1, \quad X \in \mathcal{DN}^n\right\}.$$

- We have $\ell_1(Q) \leq \ell_2(Q) \leq \nu(Q)$ [Bomze et al., 2008].
- $\ell_1(Q)$ can be computed in $O(n^2)$ time whereas $\ell_2(Q)$ requires solving a computationally expensive semidefinite program.
- For other lower bounds, see, e.g., [Nowak, 1999], [Bomze and de Klerk, 2002], [Anstreicher and Burer, 2005], and [Bomze et al., 2008] for a comparison.
- In (MILP1), we can use $M_j := \max_{i=1,\ldots,n} Q_{ij} - \ell, \; j = 1, \ldots, n$, where $\ell \in \{\ell_1(Q), \ell_2(Q)\}$.

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,

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

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),

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

A Min-Max Based MILP Formulation

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

A Min-Max Based MILP Formulation

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

Remark

Given an instance of (StQP), (MILP2) is an equivalent reformulation of (StQP) if $K_j \geq 1, \ j = 1, \ldots, n$ *and*

$$U_j \geq M_j = \max_{i=1,\ldots,n} Q_{ij} - \ell, \quad j = 1, \ldots, n,$$

where 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 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 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 and Valid Inequalities

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

Convexity Graph and Valid Inequalities

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

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 and Valid Inequalities

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

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 a globally optimal solution $x^* \in \Delta_n$ of (StQP) such that the vertices corresponding to $\mathcal{P}(x^*)$ form a clique in the convexity graph $G = (V, E)$ of Q (or, equivalently, a stable set in the complement of G).

Convexity Graph and Valid Inequalities

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

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 a globally optimal solution $x^* \in \Delta_n$ of (StQP) such that the vertices corresponding to $\mathcal{P}(x^*)$ form a clique in the convexity graph $G = (V, E)$ of Q (or, equivalently, a stable set in the complement of G).

Theorem

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

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

Outline

Computational Setup I

- Two MILP formulations: KKT Based (MILP1) and Min-Max Based (MILP2)

Computational Setup I

- Two MILP formulations: KKT Based (MILP1) and Min-Max Based (MILP2)
- Two lower bounds: $\ell_1(Q)$ and $\ell_2(Q)$

Computational Setup I

- Two MILP formulations: KKT Based (MILP1) and Min-Max Based (MILP2)
- Two lower bounds: $\ell_1(Q)$ and $\ell_2(Q)$
- Two choices: All valid inequalities vs no valid inequalities

Computational Setup I

- Two MILP formulations: KKT Based (MILP1) and Min-Max Based (MILP2)
- Two lower bounds: $\ell_1(Q)$ and $\ell_2(Q)$
- Two choices: All valid inequalities vs no valid inequalities
- A total of eight variants denoted by MILPα-Lβ (no valid inequalities) and MILPα-Lβ-VI (with valid inequalities), where $\alpha \in \{1, 2\}$ corresponds to (MILP1) and (MILP2), and $\beta \in \{1, 2\}$ corresponds to $\ell_1(Q)$ and $\ell_2(Q)$.

Computational Setup I

- Two MILP formulations: KKT Based (MILP1) and Min-Max Based (MILP2)
- Two lower bounds: $\ell_1(Q)$ and $\ell_2(Q)$
- Two choices: All valid inequalities vs no valid inequalities
- A total of eight variants denoted by MILPα-Lβ (no valid inequalities) and MILPα-Lβ-VI (with valid inequalities), where $\alpha \in \{1, 2\}$ corresponds to (MILP1) and (MILP2), and $\beta \in \{1, 2\}$ corresponds to $\ell_1(Q)$ and $\ell_2(Q)$.
- We add the following bound constraints to (MILP1) and (MILP2):

$$\ell \leq \left\{ \begin{array}{c} \lambda \\ \alpha \end{array} \right\} \leq \min_{k=1,\ldots,n} Q_{kk}.$$

Computational Setup II

- We compare the performances of our MILP formulations with three other global solution approaches:
  (i) MILP formulation of [Xia, Vera, Zuluaga, 2015] (denoted by QP-IP)
  (ii) Nonconvex QP solver of CPLEX (denoted by CPLEX QP)
  (iii) NLP solver BARON (denoted by BARON)

Computational Setup II

- We compare the performances of our MILP formulations with three other global solution approaches:
  - (i) MILP formulation of [Xia, Vera, Zuluaga, 2015] (denoted by QP-IP)
  - (ii) Nonconvex QP solver of CPLEX (denoted by CPLEX QP)
  - (iii) NLP solver BARON (denoted by BARON)
- CPLEX (version 12.8.0) with the CPLEX Class API provided in CPLEX for MATLAB Toolbox

Computational Setup II

- We compare the performances of our MILP formulations with three other global solution approaches:
  - (i) MILP formulation of [Xia, Vera, Zuluaga, 2015] (denoted by QP-IP)
  - (ii) Nonconvex QP solver of CPLEX (denoted by CPLEX QP)
  - (iii) NLP solver BARON (denoted by BARON)
- CPLEX (version 12.8.0) with the CPLEX Class API provided in CPLEX for MATLAB Toolbox
- BARON (version 17.4.1)

Computational Setup II

- We compare the performances of our MILP formulations with three other global solution approaches:
  - (i) MILP formulation of [Xia, Vera, Zuluaga, 2015] (denoted by QP-IP)
  - (ii) Nonconvex QP solver of CPLEX (denoted by CPLEX QP)
  - (iii) NLP solver BARON (denoted by BARON)
- CPLEX (version 12.8.0) with the CPLEX Class API provided in CPLEX for MATLAB Toolbox
- BARON (version 17.4.1)
- MOSEK (version 8.1.0.49)

Computational Setup II

- We compare the performances of our MILP formulations with three other global solution approaches:
  - (i) MILP formulation of [Xia, Vera, Zuluaga, 2015] (denoted by QP-IP)
  - (ii) Nonconvex QP solver of CPLEX (denoted by CPLEX QP)
  - (iii) NLP solver BARON (denoted by BARON)
- CPLEX (version 12.8.0) with the CPLEX Class API provided in CPLEX for MATLAB Toolbox
- BARON (version 17.4.1)
- MOSEK (version 8.1.0.49)
- Time limit of 3600 seconds

Computational Setup II

- We compare the performances of our MILP formulations with three other global solution approaches:
    - (i) MILP formulation of [Xia, Vera, Zuluaga, 2015] (denoted by QP-IP)
    - (ii) Nonconvex QP solver of CPLEX (denoted by CPLEX QP)
    - (iii) NLP solver BARON (denoted by BARON)
- CPLEX (version 12.8.0) with the CPLEX Class API provided in CPLEX for MATLAB Toolbox
- BARON (version 17.4.1)
- MOSEK (version 8.1.0.49)
- Time limit of 3600 seconds
- 64-bit HP workstation with 24 threads (2 sockets, 6 cores per socket, 2 threads per core) running Ubuntu Linux with 48 GB of RAM and Intel Xeon CPU E5-2667 processors with a clock speed of 2.90 GHz

Set of Instances

We conducted extensive experiments on the following set of instances from the literature:

(i) *BLST Instances* [Bonami et al., 2016]: Randomly generated instances with $n \in \{30, 50\}$.

(ii) *ST Instances* [Scozzari and Tardella, 2008]: Randomly generated instances with a prespecified convexity graph density with $n \in \{100, 200, 500, 1000\}$.

(iii) *DIMACS Instances*: Subset of maximum clique instances from the DIMACS collection with $n \in [28, 300]$.

(iv) *BSU Instances* [Bomze et al., 2018]: Specifically constructed hard instances with $n \in [5, 24]$ and the number of strict local minimizers between $(1.38)^n$ and $(1.49)^n$.

Summary

Instance Family	Number of Instances	n	δ
BLST30	134	30	[0.10,1.00]
BLST50	138	50	[0.11,1.00]
ST100	24	100	[0.25,0.90]
ST200	18	200	[0.25,0.75]
ST500	11	500	[0.25,0.50]
ST1000	1	1000	0.25
DIMACS1	8	[28,171]	[.35,.93]
DIMACS2	22	[200,300]	[.08,.97]
BSU	20	[5,24]	[.6,1]

Table 1: Summary of Instances

- δ denotes the density of the convexity graph.

Summary

Instance Family	Number of Instances	n	δ
BLST30	134	30	[0.10,1.00]
BLST50	138	50	[0.11,1.00]
ST100	24	100	[0.25,0.90]
ST200	18	200	[0.25,0.75]
ST500	11	500	[0.25,0.50]
ST1000	1	1000	0.25
DIMACS1	8	[28,171]	[.35,.93]
DIMACS2	22	[200,300]	[.08,.97]
BSU	20	[5,24]	[.6,1]

Table 1: Summary of Instances

- δ denotes the density of the convexity graph.
- 376 instances in total

BLST30 Instances

	Total Time	OPT	Time Limit	Average Gap
MILP1-L1	26.97	**134**	0	–
MILP1-L1-VI	30.71	**134**	0	–
MILP2-L1	30.50	**134**	0	–
MILP2-L1-VI	33.80	**134**	0	–
MILP1-L2	34.67	**134**	0	–
MILP1-L2-VI	**24.10**	**134**	0	–
MILP2-L2	25.65	**134**	0	–
MILP2-L2-VI	29.28	**134**	0	–
QP-IP	38.10	**134**	0	–
CPLEX QP	39370.66	124	10	6.14%
BARON	188005.68	96	38	115.91%
DNN	14.33	134	–	–

Table 2: Performance on BLST30 (134 instances; $n = 30$)

BLST50 Instances

	Total Time	OPT	Time Limit	Average Gap
MILP1-L1	50.30	**138**	0	–
MILP1-L1-VI	50.46	**138**	0	–
MILP2-L1	49.91	**138**	0	–
MILP2-L1-VI	55.54	**138**	0	–
MILP1-L2	45.06	**138**	0	–
MILP1-L2-VI	41.32	**138**	0	–
MILP2-L2	**30.48**	**138**	0	–
MILP2-L2-VI	35.25	**138**	0	–
QP-IP	85.85	**138**	0	–
CPLEX QP	77072.98	124	14	13.74%
BARON	431110.51	19	119	246.14%
DNN	99.44	138	–	–

Table 3: Performance on BLST50 (138 instances; $n = 50$)

ST 100 Instances

	Total Time	OPT	Time Limit	Average Gap
MILP1-L1	38.21	**24**	0	–
MILP1-L1-VI	52.95	**24**	0	–
MILP2-L1	20.78	**24**	0	–
MILP2-L1-VI	28.16	**24**	0	–
MILP1-L2	44.22	**24**	0	–
MILP1-L2-VI	63.88	**24**	0	–
MILP2-L2	**10.84**	**24**	0	–
MILP2-L2-VI	16.87	**24**	0	–
QP-IP	135.01	**24**	0	–
CPLEX QP	70894.32	6	18	38.18%
BARON	86404.50	0	24	1131.58%
DNN	388.39	24	–	–

Table 4: Performance on ST100 (24 instances; $n = 100$)

ST200 Instances

	Total Time	OPT	Time Limit	Average Gap
MILP1-L1	576.13	**18**	0	–
MILP1-L1-VI	512.49	**18**	0	–
MILP2-L1	189.45	**18**	0	–
MILP2-L1-VI	226.08	**18**	0	–
MILP1-L2	463.57	**18**	0	–
MILP1-L2-VI	417.84	**18**	0	–
MILP2-L2	**37.96**	**18**	0	–
MILP2-L2-VI	124.83	**18**	0	–
QP-IP	1197.43	**18**	0	–
CPLEX QP	64804.65	0	18	44.63%
BARON	64807.69	0	18	4249.18%
DNN	7729.80	18	–	–

Table 5: Performance on ST200 (18 instances; $n = 200$)

ST100 & ST 200 Instances

	Total Time	OPT	Time Limit	Average Gap
MILP1-L1	38.21	**24**	0	–
MILP1-L1-VI	52.95	**24**	0	–
MILP2-L1	20.78	**24**	0	–
MILP2-L1-VI	28.16	**24**	0	–
MILP1-L2	44.22	**24**	0	–
MILP1-L2-VI	63.88	**24**	0	–
MILP2-L2	**10.84**	**24**	0	–
MILP2-L2-VI	16.87	**24**	0	–
QP-IP	135.01	**24**	0	–
CPLEX QP	70894.32	6	18	38.18%
BARON	86404.50	0	24	1131.58%
DNN	388.39	24	–	–

Table 6: Performance on ST100 (24 instances; $n = 100$) and ST200 (18 instances; $n = 200$)

ST500 Instances

	Total Time	OPT	Time Limit	Average Gap
MILP1-L1	6583.05	**11**	0	–
MILP1-L1-VI	8960.10	**11**	0	–
MILP2-L1	**2519.94**	**11**	0	–
MILP2-L1-VI	3802.81	**11**	0	–
QP-IP	20291.34	6	5	6.97%
CPLEX QP	39603.91	0	11	44660285.88%
BARON	39600.13	0	11	11376.00%

Table 7: Performance on ST500 (11 instances; $n = 500$)

ST1000 Instance

	Total Time	OPT	Time Limit	Average Gap
MILP1-L1	3600.31	0	1	7.63%
MILP1-L1-VI	3609.65	0	1	46.74%
MILP2-L1	**2100.22**	**1**	0	–
MILP2-L1-VI	3600.18	0	1	11.59%
QP-IP	3607.88	0	1	22.05%
CPLEX QP	3600.95	0	1	86423971.55%
BARON	3601.03	0	1	18020.23%

Table 8: Performance on ST1000 (1 instance; $n = 1000$)

DIMACS1 Instances

	Total Time	OPT	Time Limit	Average Gap
MILP1-L1	3697.94	7	1	72.80%
MILP1-L1-VI	52.31	**8**	0	–
MILP2-L1	3624.48	7	1	72.80%
MILP2-L1-VI	59.72	**8**	0	–
MILP1-L2	3731.06	7	1	9.46%
MILP1-L2-VI	140.79	**8**	0	–
MILP2-L2	3616.89	7	1	9.46%
MILP2-L2-VI	**51.82**	**8**	0	–
QP-IP	4863.33	7	1	100.00%
CPLEX QP	29449.29	0	8	61.79%
BARON	28804.96	0	8	83.13%
ILP	**16.17**	**8**	0	–
DNN	356.74	8	–	–

Table 9: Performance on DIMACS1 (8 instances; $n \in [28, 171]$)

DIMACS2 Instances

	Total Time	OPT	Time Limit	Average Gap
MILP1-L1	57676.31	7	15	81.91%
MILP1-L1-VI	**21463.40**	**18**	4	71.60%
MILP2-L1	55529.78	7	15	82.38%
MILP2-L1-VI	25214.24	17	5	79.65%
MILP1-L2	56047.95	7	15	18.29%
MILP1-L2-VI	27944.05	17	5	9.40%
MILP2-L2	55127.62	8	14	20.60%
MILP2-L2-VI	30903.19	15	7	**7.65%**
QP-IP	61884.01	5	17	99.98%
CPLEX QP	77391.14	1	21	84.68%
BARON	75616.99	1	21	92.79%
ILP	**4672.59**	**21**	1	14.07%
DNN	25071.78	22	–	–

Table 10: Performance on DIMACS2 (22 instances; $n \in [200, 300]$)

BSU Instances

	Total Time	OPT	Time Limit	Average Gap
MILP1-L1	6.75	**20**	0	–
MILP1-L1-VI	5.99	**20**	0	–
MILP2-L1	8.77	**20**	0	–
MILP2-L1-VI	7.39	**20**	0	–
MILP1-L2	7.02	**20**	0	–
MILP1-L2-VI	**4.92**	**20**	0	–
MILP2-L2	7.04	**20**	0	–
MILP2-L2-VI	7.45	**20**	0	–
QP-IP	11.96	**20**	0	–
CPLEX QP	63194.15	5	15	14.82%
BARON	58022.22	5	15	54.53%
DNN	**0.51**	20	–	–

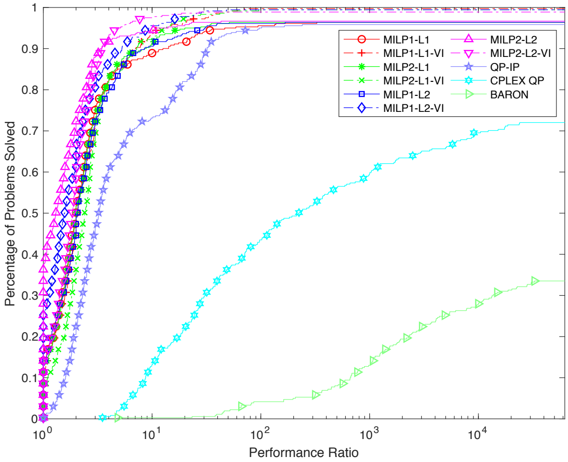
Table 11: Performance on BSU (20 instances; $n \in [5, 24]$)

Overall Comparison

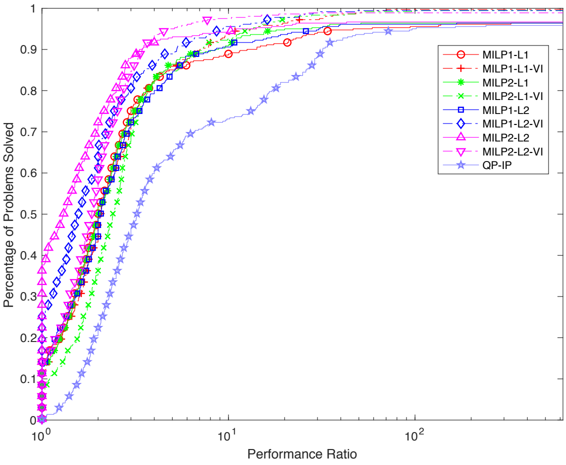
	IS1			IS2		
	Total Time	OPT	Time Limit	Total Time	OPT	Time Limit
MILP1-L1	62072.61	348	16	10183.36	11	1
MILP1-L1-VI	**22168.31**	**360**	4	12569.75	11	1
MILP2-L1	59453.67	348	16	**4620.16**	**12**	0
MILP2-L1-VI	25624.93	359	5	7402.99	11	1
MILP1-L2	60373.55	348	16	–	–	–
MILP1-L2-VI	28636.90	358	5	–	–	–
MILP2-L2	58846.48	349	15	–	–	–
MILP2-L2-VI	31168.69	357	7	–	–	–
QP-IP	68215.69	346	18	23899.22	6	6
CPLEX QP	422177.19	260	104	43204.86	0	**12**
BARON	932772.55	121	243	43201.16	0	**12**
DNN	33660.99	364	–	–	–	–

Table 12: Overall performance comparison (IS1: 364 instances; $n \in [5, 300]$; IS2: 12 instances; $n \in [500, 1000]$)

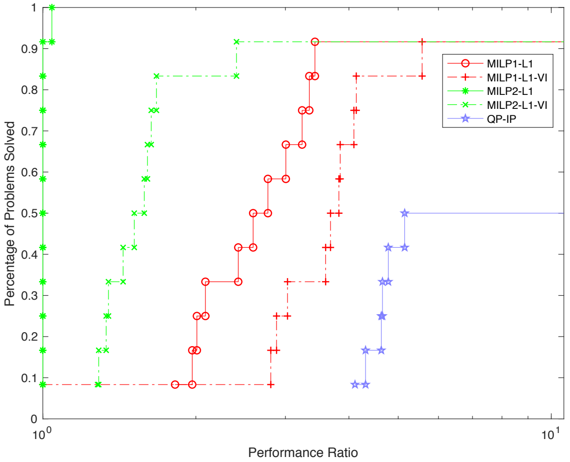
Performance Profile on IS1



Performance Profile on IS1 (w/o CPLEX QP and BARON)



Performance Profile on IS2 (w/o CPLEX QP and BARON)



Outline

Concluding Remarks

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

Concluding Remarks

- We proposed solving standard quadratic programs by using two alternative MILP reformulations.
- We considered two different choices for the big-M parameters.

Concluding Remarks

- We proposed solving standard quadratic programs by using two alternative MILP reformulations.
- We considered two different choices for the big-M parameters.
- We proposed valid inequalities for our MILP formulations.

Concluding Remarks

- We proposed solving standard quadratic programs by using two alternative MILP reformulations.
- We considered two different choices for the big-M parameters.
- We proposed valid inequalities for our MILP formulations.
- Promising computational results on a fairly large set of instances with different characteristics.

Concluding Remarks

- We proposed solving standard quadratic programs by using two alternative MILP reformulations.
- We considered two different choices for the big-M parameters.
- We proposed valid inequalities for our MILP formulations.
- Promising computational results on a fairly large set of instances with different characteristics.
- Faster computation of $\ell_2(Q)$ using proximal gradient methods?

Concluding Remarks

- We proposed solving standard quadratic programs by using two alternative MILP reformulations.
- We considered two different choices for the big-M parameters.
- We proposed valid inequalities for our MILP formulations.
- Promising computational results on a fairly large set of instances with different characteristics.
- Faster computation of $\ell_2(Q)$ using proximal gradient methods?
- Alternative tighter cheap lower bounds? (e,g,. LP based bound of [Liuzzi et al., 2017])

Concluding Remarks

- We proposed solving standard quadratic programs by using two alternative MILP reformulations.
- We considered two different choices for the big-M parameters.
- We proposed valid inequalities for our MILP formulations.
- Promising computational results on a fairly large set of instances with different characteristics.
- Faster computation of $\ell_2(Q)$ using proximal gradient methods?
- Alternative tighter cheap lower bounds? (e,g,. LP based bound of [Liuzzi et al., 2017])
- Tight MILP formulations for other classes of quadratic programs?

Acknowledgements

- Irene Klein, Vera Lehmwald, Markus Leitner, Ivana Ljubic, and Werner Schachinger
- TÜBİTAK-BİDEB 2219 International Postdoctoral Research Scholarship Program
- School of Mathematics, University of Edinburgh
- Paper available on arXiv and Optimization Online

HAPPY BIRTHDAY, MANUEL!