ON SOLVING THE QUADRATIC SHORTEST PATH PROBLEM

Renata Sotirov & Hao Hu

Tilburg University, The Netherlands
The Quadratic Shortest Path Problem (QSPP)

- $G = (V, A)$ ... a directed graph
  - $V$ ... vertex set, $|V| = n$
  - $A$ ... arc set, $|A| = m$
- $s, t \in V$ ... two distinguished vertices
The Quadratic Shortest Path Problem (QSPP)

- $G = (V, A)$ … a directed graph
  - $V$ … vertex set, $|V| = n$
  - $A$ … arc set, $|A| = m$
- $s, t \in V$ … two distinguished vertices

The shortest path problem (SPP)

Find a path between $s$ and $t$ s.t. the sum of costs of arcs on the path is minimized.
The Quadratic Shortest Path Problem (QSPP)

- $G = (V, A)$ ... a directed graph
  - $V$ ... vertex set, $|V| = n$
  - $A$ ... arc set, $|A| = m$
- $s, t \in V$ ... two distinguished vertices

The shortest path problem (SPP)
Find a path between $s$ and $t$ s.t. the sum of costs of arcs on the path is minimized.

The quadratic shortest path problem (QSPP)
Find a path between $s$ and $t$ s.t. the sum of costs of arcs, and the sum of interaction costs over all pairs of arcs on the path is minimized.
The Shortest Path Problem (SPP)

- find a path $s \rightarrow t$ s.t. the linear cost of the path is minimized
The Shortest Path Problem (SPP)

- find a path \( s \rightarrow t \) s.t. the **linear cost** of the path is minimized

- \( c \in \mathbb{R}^m \) ... vector of **arc costs**
The Shortest Path Problem (SPP)

- find a path $s \rightarrow t$ s.t. the linear cost of the path is minimized
- $c \in \mathbb{R}^m_+$ . . . vector of arc costs
- $x \in \{0, 1\}^m$ . . . the characteristic vector of a path $P$

$$x_{ij} = \begin{cases} 1 & \text{if } (i, j) \in A \text{ is on the } s-t \text{ path} \\ 0 & \text{otherwise} \end{cases}$$
The Shortest Path Problem (SPP)

- find a path $s \rightarrow t$ s.t. the linear cost of the path is minimized
- $c \in \mathbb{R}_+^m$ ... vector of arc costs
- $x \in \{0, 1\}^m$ ... the characteristic vector of a path $P$
- $x_{ij} = \begin{cases} 1 & \text{if } (i, j) \in A \text{ is on the } s-t \text{ path} \\ 0 & \text{otherwise} \end{cases}$
- $I \in \mathbb{R}^{n\times m}$ ... the incidence matrix of $G$
The Shortest Path Problem (SPP)

- find a path \( s \to t \) s.t. the linear cost of the path is minimized
- \( c \in \mathbb{R}_+^m \) ... vector of arc costs
- \( x \in \{0,1\}^m \) ... the characteristic vector of a path \( P \)
  \[
  x_{ij} = \begin{cases} 
  1 & \text{if } (i,j) \in A \text{ is on the } s-t \text{ path} \\
  0 & \text{otherwise} 
  \end{cases}
  \]
- \( \mathcal{I} \in \mathbb{R}^{n \times m} \) ... the incidence matrix of \( G \)

\[
\begin{align*}
\min \quad & c^T x \\
\text{(SPP)} \quad & \mathcal{I} x = b \\
\text{s.t.} \quad & x \in \{0,1\}^m \\
\end{align*}
\]

where \( b_s = 1, \ b_t = -1 \) and \( b_i = 0 \) if \( i \in V \setminus \{s, t\} \)
The Quadratic Shortest Path Problem (QSPP)

- find a path $s \rightarrow t$ s.t. the quadratic cost of the path is minimized
The Quadratic Shortest Path Problem (QSPP)

- find a path $s \rightarrow t$ s.t. the quadratic cost of the path is minimized
- $Q = Q^T \in \mathbb{R}^{m \times m}_+$ ... matrix of interaction costs
The Quadratic Shortest Path Problem (QSPP)

- find a path $s \to t$ s.t. the quadratic cost of the path is minimized
- $Q = Q^T \in \mathbb{R}_{+}^{m \times m}$ ... matrix of interaction costs
- $x \in \{0, 1\}^m$ ... the characteristic vector of a path $P$

$$x_{ij} = \begin{cases} 1 & \text{if } (i, j) \in A \text{ is on the } s-t \text{ path} \\ 0 & \text{otherwise} \end{cases}$$
The Quadratic Shortest Path Problem (QSPP)

- find a path $s \rightarrow t$ s.t. the quadratic cost of the path is minimized
- $Q = Q^T \in \mathbb{R}_+^{m \times m}$ ... matrix of interaction costs
- $x \in \{0, 1\}^m$ ... the characteristic vector of a path $P$

$$
\begin{align*}
x_{ij} &= \begin{cases} 
1 & \text{if } (i, j) \in A \text{ is on the } s-t \text{ path} \\
0 & \text{otherwise}
\end{cases} \\
\end{align*}
$$

$$
\begin{align*}
\min_{x} & \quad x^T Q x \\
\text{s.t.} & \quad \mathcal{I} x = b \\
& \quad x \in \{0, 1\}^m
\end{align*}
$$
The QSPP

The QSPP . . .

- is NP-hard


  Hu, Sotirov, Special cases of the QSPP. arXiv:1611.07682
The QSPP

The QSPP . . .

- is NP-hard


  Hu, Sotirov, Special cases of the QSPP. arXiv:1611.07682

- applications:
  - transportation:
    - of hazardous materials
    - unloading and reloading goods at junctions
  - telecommunication networks
  - satellite network design
  - energy distribution networks
  - route planning
  - . . .
Poly-time solvable cases of the QSPP
Linearizable QSPP

A QSPP instance is given by: $G, s, t \in V$, matrix $Q$ where

- $x^T Q x$ ... the quadratic cost of the $s$-$t$ path $P_x$
Linearizable QSPP

A QSPP instance is given by: \( G, s, t \in V \), matrix \( Q \) where

- \( x^T Q x \) . . . the quadratic cost of the \( s-t \) path \( P_x \)

Linearizable QSPP

A QSPP instance is linearizable if there exists a cost vector \( c \) s.t.

\[ x^T Q x = x^T c \]

for every \( s-t \) path in \( G \)

We call \( c \) the linearization vector of \( Q \).
Linearizable QSPP

A QSPP instance is given by: $G, s, t \in V$, matrix $Q$ where
- $x^T Q x \ldots$ the quadratic cost of the $s$-$t$ path $P_x$

Linearizable QSPP

A QSPP instance is linearizable if there exists a cost vector $c$ s.t.

$$x^T Q x = x^T c$$

for every $s$-$t$ path in $G$

We call $c$ the linearization vector of $Q$.

- An instance may be linearizable if:
  - $Q$ has special properties
  - $G$ has special structure
Examples of Linearizable QSPP

- the adjacent QSPP:
  
  \[ q_{e,f} = \begin{cases} 
  \geq 0 & \text{if } e, f \in A \text{ are adjacent} \\
  = 0 & \text{otherwise} 
  \end{cases} \]

⇒ linearizable if \( G \) is directed acyclic graph

Examples of Linearizable QSPP

- the adjacent QSPP:

\[ q_{e,f} = \begin{cases} 
\geq 0 & \text{if } e, f \in A \text{ are adjacent} \\
= 0 & \text{otherwise} 
\end{cases} \]

⇒ linearizable if \( G \) is directed acyclic graph


⇒ in general NP-hard

Hu, S. Special cases of the QSPP, Preprint, 2016.

Examples of Linearizable QSPP

- the adjacent QSPP:

\[
q_{e,f} = \begin{cases} 
\geq 0 & \text{if } e, f \in A \text{ are adjacent} \\
= 0 & \text{otherwise}
\end{cases}
\]

⇒ linearizable if \( G \) is directed acyclic graph

- Rostami, Malucelli, Frey, Buchheim. On the QSPP.

⇒ in general NP-hard

- Hu, S. Special cases of the QSPP, Preprint, 2016.

- \( Q \) is generated by some \( w \in \mathbb{R}^m_+ \):

\[
q_{e,f} = w_e + w_f \quad e, f \in A
\]

and every \( s-t \) path in \( G \) has the same length
Graphs whose every $s$-$t$ path has constant length?

- The directed grid graph $G_{pq} = (V, A)$ where
  
  $$V = \{v_{i,j} \mid 1 \leq i \leq p, 1 \leq j \leq q\}$$

  $$A = \{(v_{i,j}, v_{i+1,j}) \mid 1 \leq i \leq p - 1, 1 \leq j \leq q\}$$

  $$\cup \{(v_{i,j}, v_{i,j+1}) \mid 1 \leq i \leq p, 1 \leq j \leq q - 1\}$$

  and every $v_{1,1}$-$v_{p,q}$ path has length $p + q - 2$. 

Figure: The directed grid graph $G_{3,4}$, variants of $G_{pq}$, hypercube graph, etc.
Graphs whose every \( s-t \) path has constant length?

- The directed grid graph \( G_{pq} = (V, A) \) where
  \[
  V = \{v_{i,j} \mid 1 \leq i \leq p, 1 \leq j \leq q\}
  \]
  \[
  A = \{(v_{i,j}, v_{i+1,j}) \mid 1 \leq i \leq p-1, 1 \leq j \leq q\}
  \]
  \[
  \cup \{(v_{i,j}, v_{i,j+1}) \mid 1 \leq i \leq p, 1 \leq j \leq q-1\}
  \]
  and every \( v_{1,1} - v_{p,q} \) path has length \( p + q - 2 \).

- \( |V| = pq \) and \( |A| = 2pq - p - q \)

\[\text{Figure : The directed grid graph } G_{3,4}\]
Graphs whose every $s$-$t$ path has constant length?

- The directed grid graph $G_{pq} = (V, A)$ where
  \[ V = \{ v_{i,j} \mid 1 \leq i \leq p, 1 \leq j \leq q \} \]
  \[ A = \{ (v_{i,j}, v_{i+1,j}) \mid 1 \leq i \leq p-1, 1 \leq j \leq q \} \cup \{ (v_{i,j}, v_{i,j+1}) \mid 1 \leq i \leq p, 1 \leq j \leq q-1 \} \]
  and every $v_{1,1}$-$v_{p,q}$ path has length $p + q - 2$.

- $|V| = pq$ and $|A| = 2pq - p - q$

variants of $G_{pq}$, hypercube graph, etc.
The QSPP on the directed grid graph

- consider the QSPP with no restriction on $Q$, and $G_{p,q}$
The QSPP on the directed grid graph

- consider the QSPP with no restriction on $Q$, and $G_{p,q}$

Thm (Hu & S.)

The algorithm \textsc{Linearize-grid} determines if a QSPP instance on $G_{p,q}$ is linearizable, and if so it constructs its linearization vector in $O(p^3 q^2 + p^2 q^3)$ time.
The QSPP on the directed grid graph

- consider the QSPP with no restriction on $Q$, and $G_{p,q}$

**Thm (Hu & S.)**

The algorithm `LINEARIZE-GRID` determines if a QSPP instance on $G_{p,q}$ is linearizable, and if so it constructs its linearization vector in $O(p^3q^2 + p^2q^3)$ time.

**Proof.** (sketch)

- Show that $\mathcal{I}_1 = (G_{p,q}, v_{1,1}, v_{p,q}, Q)$ is linearizable iff $\mathcal{I}_2 = (G_{p,q}, v_{1,1}, v_{p-1,q}, Q)$ and $\mathcal{I}_3 = (G_{p,q}, v_{1,1}, v_{p,q-1}, Q)$ are linearizable.
The directed grid graph

Thm (Hu & S.)

The algorithm **Linearize-grid** determines if a QSPP instance on $G_{p,q}$ is linearizable, and if so it constructs its linearization vector in $O(p^3q^2 + p^2q^3)$ time.

**Proof.** (sketch)

- Show that $I_1 = (G_{p,q}, v_{1,1}, v_{p,q}, Q)$ is linearizable iff
  $I_2 = (G_{p,q}, v_{1,1}, v_{p-1,q}, Q)$ and
  $I_3 = (G_{p,q}, v_{1,1}, v_{p,q-1}, Q)$ are linearizable.

- Show that for $I_1 = (G_{p,q}, v_{1,1}, v_{p,q}, Q)$ only $(p - 1)(q - 1) + 1$ paths are important for linearization,
The directed grid graph

Thm (Hu & S.)

The algorithm **LINEARIZE-GRID** determines if a QSPP instance on $G_{p,q}$ is linearizable, and if so it constructs its linearization vector in $O(p^3q^2 + p^2q^3)$ time.

**Proof.** (sketch)

- Show that $I_1 = (G_{p,q}, v_{1,1}, v_{p,q}, Q)$ is *linearizable* iff
  
  $I_2 = (G_{p,q}, v_{1,1}, v_{p-1,q}, Q)$ and
  
  $I_3 = (G_{p,q}, v_{1,1}, v_{p,q-1}, Q)$ are
  
  *linearizable*.

- Show that for $I_1 = (G_{p,q}, v_{1,1}, v_{p,q}, Q)$ only $(p - 1)(q - 1) + 1$ paths are important for linearization, and similar for $I_2$ and $I_3$, etc.
The directed grid graph

**Thm (Hu & S.)**

The algorithm `LINEARIZE-GRID` determines if a QSPP instance on $G_{p,q}$ is linearizable, and if so it constructs its linearization vector in $O(p^3q^2 + p^2q^3)$ time.

**Proof.** (sketch)

- Show that $I_1 = (G_{p,q}, v_{1,1}, v_{p,q}, Q)$ is linearizable iff
  - $I_2 = (G_{p,q}, v_{1,1}, v_{p-1,q}, Q)$ and
  - $I_3 = (G_{p,q}, v_{1,1}, v_{p,q-1}, Q)$ are linearizable.

- Show that for $I_1 = (G_{p,q}, v_{1,1}, v_{p,q}, Q)$ only $(p - 1)(q - 1) + 1$ paths are important for linearization, and similar for $I_2$ and $I_3$, etc.

- Show that all QSPP instances on $G_{2,q}$ for $q \geq 2$ are linearizable. □
The directed grid graph

- The algorithm can be adjusted for any directed acyclic graph (DAG).
The directed grid graph

- The algorithm can be adjusted for any directed acyclic graph (DAG).
- The complexity of the algorithm for a DAG with is $\mathcal{O}(|A|^4 + |V||A|^3)$.
The directed grid graph

- The algorithm can be adjusted for any directed acyclic graph (DAG).
- The complexity of the algorithm for a DAG with is $O(|A|^4 + |V||A|^3)$
- Can we use **Linearize-DAG** to solve difficult QSPP instances?
Exploiting **LINEARIZE-GRID** to derive bounds

- consider the QSPP with *no restriction on* $Q$, and $G_{p,q}$

How to compute $l_p$ bounds?

**Strategy**:
Find a linearizable $\hat{Q}$ that is “close” to the cost matrix $Q$.

**Linearize-DAG** yields the linear system:

$$B(\hat{Q}) = 0 \Rightarrow \hat{Q} \text{is linearizable}$$

the following splitting approach provides $\hat{Q}$ and its linearization vector $\hat{c}$:

$$\max \hat{Q}, \hat{c} \quad \sum_i \hat{c}_i s.t. \quad B(\hat{Q}) = 0 \quad (\hat{Q} \text{is linearizable})$$

$$C(\hat{Q}) = \hat{c}$$

$$Q - \hat{Q} \geq 0 \quad \text{(quadratic cost matrix is non-negative)}$$

there are no similar splitting approaches in the literature (!)
Exploiting **LINEARIZE-GRID** to derive bounds

- consider the QSPP with no restriction on $Q$, and $G_{p,q}$

  How to compute $l_p$ bounds?
Exploiting **LINEARIZE-GRID** to derive bounds

- consider the QSPP with no restriction on $Q$, and $G_{p,q}$

  How to compute lp bounds?

**Strategy:** Find a linearizable $\hat{Q}$ that is “close” to the cost matrix $Q$. 
Exploiting **LINEARIZE-GRID** to derive bounds

- consider the QSPP with no restriction on $Q$, and $G_{p,q}$

How to compute $l_p$ bounds?

**Strategy:** Find a linearizable $\hat{Q}$ that is “close” to the cost matrix $Q$.

- **LINEARIZE-DAG** yields the linear system: $B(\hat{Q}) = 0 \Rightarrow \hat{Q}$ is linearizable

$\text{max} \hat{Q}, \hat{c}, \sum_i \hat{c}_i s.t. B(\hat{Q}) = 0 (\hat{Q} \text{ is linearizable})$,

$C(\hat{Q}) = \hat{c}$ ($\hat{c} \leftarrow \text{linearization vector}$)

$Q - \hat{Q} \geq 0$ (quadratic cost matrix is non-negative)

There are no similar splitting approaches in the literature (!)
Exploiting **LINEARIZE-GRID** to derive bounds

- consider the QSPP with **no restriction on** $Q$, and $G_{p,q}$

**How to compute $l_p$ bounds?**

**Strategy:** Find a **linearizable** $\hat{Q}$ that is “close” to the cost matrix $Q$.

- **LINEARIZE-DAG** yields the linear system: $B(\hat{Q}) = 0 \Rightarrow \hat{Q}$ is linearizable
- the following **splitting approach** provides $\hat{Q}$ and its linearization vector $\hat{c}$:

\[
\begin{align*}
\max_{\hat{Q}, \hat{c}} & \quad \sum_i \hat{c}_i \\
\text{s.t.} & \quad B(\hat{Q}) = 0 \quad (\hat{Q} \text{ is linearizable}) \\
& \quad C(\hat{Q}) = \hat{c} \quad (\hat{c} \leftarrow \text{linearization vector}) \\
& \quad Q - \hat{Q} \geq 0 \quad (\text{quadratic cost matrix is non-negative})
\end{align*}
\]
Exploiting **LINEARIZE-GRID** to derive bounds

- consider the QSPP with **no restriction on** $Q$, and $G_{p,q}$

  How to compute lp bounds?

**Strategy:** Find a **linearizable** $\hat{Q}$ that is “close” to the cost matrix $Q$.

- **LINEARIZE-DAG** yields the linear system: $B(\hat{Q}) = 0 \Rightarrow \hat{Q}$ is linearizable

- the following **splitting approach** provides $\hat{Q}$ and its linearization vector $\hat{c}$:

  $$\max_{\hat{Q},\hat{c}} \sum_i \hat{c}_i$$

  s.t. $B(\hat{Q}) = 0$ ($\hat{Q}$ is linearizable)

  $C(\hat{Q}) = \hat{c}$ ($\hat{c} \leftarrow$ linearization vector)

  $Q - \hat{Q} \geq 0$ (quadratic cost matrix is non-negative)

- there are **no** similar splitting approaches in the literature (!)
Exploiting **LINEARIZE-GRID** to derive bounds

**Numerical results**

we compare:
Exploiting **LINEARIZE-GRID** to derive bounds

**Numerical results**

- we compare:
  - Gilmore-Lower bnd. obtained by underestimating quadratic cost of each arc
  - H&S our lower bound derived by **LINEARIZE-GRID**

<table>
<thead>
<tr>
<th>$G_{p,r}$</th>
<th>m</th>
<th>density</th>
<th>GL</th>
<th>H&amp;S</th>
<th>OPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>180</td>
<td>20%</td>
<td>0</td>
<td>116</td>
<td>136</td>
</tr>
<tr>
<td>13</td>
<td>312</td>
<td>40%</td>
<td>86</td>
<td>673</td>
<td>762</td>
</tr>
<tr>
<td>15</td>
<td>420</td>
<td>50%</td>
<td>299</td>
<td>1347</td>
<td>1438</td>
</tr>
</tbody>
</table>
Exploiting \textsc{Linearize-Grid} to derive bounds

\section*{Numerical results}

- we compare:
  - Gilmore-Lower bnd. obtained by underestimating quadratic cost of each arc
  - H&S our lower bound derived by \textsc{Linearize-Grid}

<table>
<thead>
<tr>
<th>$G_{p,p}$</th>
<th>$m$</th>
<th>density</th>
<th>GL</th>
<th>H&amp;S</th>
<th>OPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>180</td>
<td>20%</td>
<td>0</td>
<td>116</td>
<td>136</td>
</tr>
<tr>
<td>13</td>
<td>312</td>
<td>40%</td>
<td>86</td>
<td>673</td>
<td>762</td>
</tr>
<tr>
<td>15</td>
<td>420</td>
<td>50%</td>
<td>299</td>
<td>1347</td>
<td>1438</td>
</tr>
</tbody>
</table>

\ldots TO BE CONTINUED
Other approaches to solve the QSPP?
Other approaches to solve the QSPP?

⇒ Consider *semidefinite programming* (SDP)
SDP for the QSPP

- define the path polyhedron:

\[ P_{st}(G) := \{ x \in \mathbb{R}^m \mid \mathcal{I}x = b, \ 0 \leq x \leq 1 \} \]
SDP for the QSPP

- define the path polyhedron:

\[ P_{st}(G) := \{ x \in \mathbb{R}^m \mid \mathcal{I}x = b, \ 0 \leq x \leq 1 \} \]

- \( \text{OPT}_{\text{QSPP}} = \min \ \{ x^T Q x : x \in P_{st}(G) \cap \{0, 1\}^m \} \)
SDP for the QSPP

- define the path polyhedron:

\[ P_{st}(G) := \{ x \in \mathbb{R}^m \mid \mathcal{I}x = b, \ 0 \leq x \leq 1 \} \]

- \( \text{OPT}_{\text{QSPP}} = \min \ \{ x^T Q x : x \in P_{st}(G) \cap \{0, 1\}^m \} \)

- linearize objective: \( \text{tr}(x^T Q x) = \text{tr}(Qxx^T) \sim \langle Q, X \rangle \) where \( X \in S_m \)
**SDP for the QSPP**

- define the **path polyhedron**:

\[ P_{st}(G) := \{x \in \mathbb{R}^m \mid Ix = b, \ 0 \leq x \leq 1\} \]

- \( \text{OPT}_{\text{QSPP}} = \min \ \{x^TQx : x \in P_{st}(G) \cap \{0, 1\}^m\} \)

- linearize objective: \( \text{tr}(x^TQx) = \text{tr}(Qxx^T) \leadsto \langle Q, X \rangle \) where \( X \in S_m \)

- replace \( X \succeq 0, \ \text{rank}(X) = 1 \) by constraint \( X - xx^T \succeq 0, \ x=\text{diag}(X) \)
SDP for the QSPP

- define the path polyhedron:

\[ P_{st}(G) := \{ x \in \mathbb{R}^m | \mathcal{I}x = b, \ 0 \leq x \leq 1 \} \]

- \( \text{OPT}_{\text{QSPP}} = \min \ \{ x^T Q x : x \in P_{st}(G) \cap \{0, 1\}^m \} \)

- linearize objective: \( \text{tr}(x^T Q x) = \text{tr}(Q xx^T) \leadsto \langle Q, X \rangle \) where \( X \in S_m \)

- replace \( X \succeq 0, \text{rank}(X) = 1 \) by constraint \( X - xx^T \succeq 0, \ x=\text{diag}(X) \)

- the basic SDP relaxation:

\[
\begin{align*}
\text{min} & \quad \langle Q, X \rangle \\
\text{s.t.} & \quad x \in P_{st}(G) \\
& \quad \text{diag}(X) = x \\
& \quad \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0
\end{align*}
\]

(SDP\(_0\))
SDP for the QSPP

to improve $SDP_0$

- add nonnegativity constraints: $X \geq 0$
SDP for the QSPP

to improve $SDP_0$

- add nonnegativity constraints: $X \geq 0$
- add ‘squared linear’ constraints: $\langle \mathcal{I}_i \mathcal{I}_i^T, X \rangle = b_i^2$, $i = 1, \ldots, |V| - 1$

where $\mathcal{I}_i$ is the $i$th row of $\mathcal{I}$
SDP for the QSPP

to improve $SDP_0$

- **add** nonnegativity constraints: $X \geq 0$
- **add** ‘squared linear’ constraints: $\langle I_i I_i^T, X \rangle = b^2_i, \ i = 1, \ldots, |V| - 1$
  where $I_i$ is the $i$th row of $I$

the resulting SDP relaxation:

$$\begin{align*}
\text{min} \quad & \langle Q, X \rangle \\
\text{s.t.} \quad & I_s^T x = b_s \\
& \text{diag}(X) = x \\
& \langle I_i I_i^T, X \rangle = b^2_i, \ i = 1, \ldots, |V| - 1 \\
& \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0, \quad X \geq 0
\end{align*}$$

$SDP_N$ has $n + m$ equality, and $O(m^2)$ inequality constraints
On solving $SDP_N$

- consider QSPP instances on $G_{pq}$
On solving $SDP_N$

- consider QSPP instances on $G_{pq}$
- interior-point algorithm can solve $SDP_N$ for $m \leq 480$ arcs ($p, q \leq 16$)
  - $SDP_N$ provides tight bounds for most test instances with $m \leq 420$
On solving $SDP_N$

- consider QSPP instances on $G_{pq}$
- interior-point algorithm can solve $SDP_N$ for $m \leq 480$ arcs ($p, q \leq 16$)
  - $SDP_N$ provides tight bounds for most test instances with $m \leq 420$
- to solve LARGER INSTANCES we implemented:
  the Alternating Direction Method of Multipliers (ADMM)
The ADMM for $SDP_N$

- we implement a variant of ADMM for SDP from:

\[
\begin{align*}
\min & \quad \langle Q, Y \rangle \\
\text{s.t.} & \quad \text{diag}(Y) = Y_{m+1}^T Y_{m+1} = 1 \\
& \quad Y = WUW^T \\
& \quad Y \geq 0, U \succeq 0
\end{align*}
\]

The SDP relaxations $SDP_N$ and $SDP_N$ are equivalent.
The ADMM for $SDP_N$


- by using facial reduction - Borwein and Wolkowicz, 1980 - $\leadsto$ the Slater feasible version of $SDP_N$:

$$\begin{align*}
\text{min} & \quad \langle Q, Y_{1:m,1:m} \rangle \\
\text{s.t.} & \quad \text{diag}(Y) = Ye_{m+1} \\
& \quad Y_{m+1,m+1} = 1 \\
& \quad Y = WUW^T \\
& \quad Y \succeq 0, \quad U \succeq 0
\end{align*}$$

$(SDP_{NS})$

where $W \in \mathbb{R}^{m+1,m-n+2}$ and $U \in \mathcal{S}_{m-n+2}$
The ADMM for $SDP_N$


- by using facial reduction - Borwein and Wolkowicz, 1980 - ⇝ the Slater feasible version of $SDP_N$:

\[
\begin{align*}
\min & \quad \langle Q, Y_{1:m,1:m} \rangle \\
\text{s.t.} & \quad \text{diag}(Y) = Ye_{m+1} \\
& \quad Y_{m+1,m+1} = 1 \\
& \quad Y = WUW^T \\
& \quad Y \succeq 0, \quad U \succeq 0
\end{align*}
\]

$SDP_{NS}$

where $W \in \mathbb{R}^{m+1,m-n+2}$ and $U \in S_{m-n+2}$

- The SDP relaxations $SDP_N$ and $SDP_{NS}$ are equivalent.
The ADMM for \( \text{SDP}_N \)

- The augmented Lagrangian:

\[
\mathcal{L}_A(U, Y, Z) = \langle Q, Y_{1:m+1,1:m+1} \rangle + \langle Z, Y - WUW^T \rangle + \frac{\beta}{2} \| Y - WUW^T \|^2,
\]

where \( Z \in S_{m+1} \) and \( \beta > 0 \)
The ADMM for $SDP_N$

- The augmented Lagrangian:

$$\mathcal{L}_A(U, Y, Z) = \langle Q, Y_{1:m+1,1:m+1} \rangle + \langle Z, Y - WUW^T \rangle + \frac{\beta}{2} \| Y - WUW^T \|^2,$$

where $Z \in S_{m+1}$ and $\beta > 0$

- the ADMM in the $k$-th iteration:

$$U_{k+1} = \arg \min_{U \succeq 0} \mathcal{L}_A(U, Y_k, Z_k)$$

$$Y_{k+1} = \arg \min_{Y \in P} \mathcal{L}_A(U_{k+1}, Y, Z_k)$$

$$Z_{k+1} = Z_k + \gamma \beta(Y_{k+1} - WU_{k+1}W^T)$$

where $P = \{ Y \in S^n \mid \text{diag}(Y) = Ye_{m+1}, Y_{m+1,m+1} = 1, Y \geq 0 \}$
The ADMM for $SDP_N$

- the $U$-subproblem:

$$U_{k+1} = \mathcal{P}_{S_+} \left( W^T (Y^k + \frac{1}{\beta} Z^k) W \right)$$

where $\mathcal{P}_{S_+}(\cdot)$ is the projection to the cone of PSD matrices.
The ADMM for $SDP_N$

- the $U$-subproblem:

$$U_{k+1} = \mathcal{P}_{S_+} \left( W^T (Y^k + \frac{1}{\beta} Z^k) W \right)$$

where $\mathcal{P}_{S_+} (\cdot)$ is the projection to the cone of PSD matrices

- the $Y$-subproblem

$$Y_{k+1} = \begin{cases} 
\max\{0, \hat{Y}_{i,j}\} & \text{if } i < j < m + 1, \\
\max\{0, \frac{1}{3} \hat{Y}_{i,i} + \frac{2}{3} \hat{Y}_{i,m+1}\} & \text{if } i = j < m + 1, \\
\max\{0, \frac{1}{3} \hat{Y}_{i,i} + \frac{2}{3} \hat{Y}_{i,m+1}\} & \text{if } i < j = m + 1, \\
1 & \text{if } i = j = m + 1,
\end{cases}$$

where $\hat{Y} = WU_{k+1}^T W^T - \frac{1}{\beta} (Q + Z^k)$
The ADMM for $SDP_{NS}$ – practical issues
from the Lagrangean dual of $SDP_{NS}$ it follows that

$$g(Z) = \min_{Y \in P} \langle \hat{Q} + Z, Y \rangle,$$
where $Z \in \{Z \mid W^TZW \preceq 0\}$  \hspace{1cm} (★)

provides a lower bound for $SDP_{NS}$
The ADMM for $SDP_{NS}$ – practical issues

- from the Lagrangean dual of $SDP_{NS}$ it follows that
  
  $$g(Z) = \min_{Y \in P} \langle \hat{Q} + Z, Y \rangle, \text{ where } Z \in \{Z \mid W^TZW \preceq 0\}$$

  provides a lower bound for $SDP_{NS}$

- use the output of the ADMM and $(★)$ $\Rightarrow$ lower bound
The ADMM for $SDP_{NS}$ – practical issues

- from the Lagrangean dual of $SDP_{NS}$ it follows that

$$g(Z) = \min_{\hat{Q} + Z} \langle \hat{Q} + Z, Y \rangle, \text{ where } Z \in \{Z \mid W^TZW \preceq 0\} \quad \text{(★)}$$

provides a lower bound for $SDP_{NS}$

- use the output of the ADMM and (★) $\Rightarrow$ lower bound

$\Rightarrow$ one can use moderate accuracy to obtain (weaker) bound
The ADMM for $SDP_{NS}$ – practical issues

- from the Lagrangean dual of $SDP_{NS}$ it follows that

\[ g(Z) = \min_{Y \in P} \langle \hat{Q} + Z, Y \rangle, \text{ where } Z \in \{Z \mid W^TZW \preceq 0\} \quad (\star) \]

provides a lower bound for $SDP_{NS}$

- use the output of the ADMM and (\star) $\rightsquigarrow$ lower bound

\[ \Rightarrow \text{ one can use moderate accuracy to obtain (weaker) bound} \]

- weaker bounds that are computed faster are useful within a B&B framework
The ADMM for $SDP_{NS}$ – practical issues

- to improve the initial performance of the ADMM add redundant const.:
  - $Y \leq 1$ (see Oliveira et al.)
The ADMM for $SDP_{NS}$ – practical issues

- to improve the initial performance of the ADMM add redundant const.:
  - $Y \leq 1$ (see Oliveira et al.)
  - $e^T Ye = (L + 1)^2$, where $L$ is the length of the path (when applicable)
The ADMM for $SDP_{NS}$ – practical issues

- to improve the **initial performance** of the ADMM add **redundant** const.:
  - $Y \leq 1$ (see Oliveira et al.)
  - $e^T Ye = (L + 1)^2$, where $L$ is the length of the path (when applicable)
- the effect of the redundant constraints is **non-beneficial** in the long run
The ADMM for $SDP_{NS}$ – practical issues

- to improve the initial performance of the ADMM add redundant const.:
  - $Y \leq 1$ (see Oliveira et al.)
  - $e^T Y e = (L + 1)^2$, where $L$ is the length of the path (when applicable)
- the effect of the redundant constraints in non-beneficial in the long run
- however, effective within a B&B framework (!)
Numerical results - summary

for the QSPP instances on the GRID GRAPHS:

- ADMM computes $SDP_{NS}$ bound for instances with $m \leq 480$ in $\leq 1$ min.
for the QSPP instances on the grid graphs:

- ADMM computes $SDP_{NS}$ bound for instances with $m \leq 480$ in $\leq 1$ min
- B&B algorithm that computes $SDP_N$ with ADMM in each node of the tree:
  - solves the QSPP with $m \leq 760$ within 3 min
  - solves the QSPP with $m = 1200$ within 30 min
- CplexQP can not provide an opt. solution for $m = 420$ within 10 000 s
Numerical results - summary

for the QSPP instances on the **GRID GRAPHS**:

- **ADMM** computes $SDP_{NS}$ bound for instances with $m \leq 480$ in $\leq 1$ min
- **B&B** algorithm that computes $SDP_N$ with **ADMM** in each node of the tree:
  - solves the QSPP with $m \leq 760$ within 3 min
  - solves the QSPP with $m = 1200$ within 30 min
- **CplexQP** can **not** provide an opt. solution for $m = 420$ within 10 000 s
Numerical results

we also tested:

- $SDP_{NS}$ and ADMM on different graphs, with

- dense data matrices
- sparse data matrices
- $k$-partite graphs

Question: Other splitting approaches for solving SDPs?
Numerical results

we also tested:

- $SDP_{NS}$ and ADMM on different graphs, with
  - sparse data matrices
  - dense data matrices
  - reload cost data used in transportation
  - $k$-partite graphs
Numerical results

we also tested:

- SDP<sub>NS</sub> and ADMM on different graphs, with
  - sparse data matrices
  - dense data matrices
  - reload cost data used in transportation
  - k-partite graphs

- largest solved instances in < 60 min has \( m = 2646 \) arcs
Numerical results

we also tested:

- $SDP_{NS}$ and ADMM on different graphs, with
  - sparse data matrices
  - dense data matrices
  - reload cost data used in transportation
  - $k$-partite graphs

- largest solved instances in $< 60$ min has $m = 2646$ arcs

QUESTION:

Other splitting approaches for solving SDPs?
THANK YOU FOR YOUR ATTENTION