Logical Benders Decomposition for Quadratic Programs with Complementarity Constraints and Binary Variables

John E. Mitchell
Francisco Jara-Moroni
Jong-Shi Pang
Andreas Wächter

Rensselaer Polytechnic Institute
Northwestern University
University of Southern California

EUROPT, Montreal
July 12, 2017

1Supported by AFOSR and NSF.
1. **Introduction**
   - QPCC
   - Simplified Method

2. **Master Problem**
   - How to select $p$?

3. **Cut Strengthening**
   - $\ell_1$-norm sparsification
   - Tree guided sparsification
   - Hybrid Sparsification

4. **Numerical Results**

5. **Extensions**
   - Binary Variables

6. **Summary & Future Research**
Outline

1. Introduction
   - QPCC
   - Simplified Method

2. Master Problem
   - How to select $p$?

3. Cut Strengthening
   - $l_1$-norm sparsification
   - Tree guided sparsification
   - Hybrid Sparsification

4. Numerical Results

5. Extensions
   - Binary Variables

6. Summary & Future Research
Complementarity constraints and binary variables

We are interested in problems with both complementary variables and binary variables. Eg: bilevel programs with binary upper level decision variables. Complementarity arises from KKT optimality conditions for the lower level problem.

For example:
A facility location problem, with multiple competing shippers.

Previously: we’ve developed logical Benders approaches to linear and quadratic programs with complementarity constraints. In this talk: Extend to also handle binary variables.

The emphasis in this talk will be on improved methods for handling the complementarity constraints.
Biofuel supply chain design (Bai, Ouyang, Pang)

Possible locations for fuel processing.
Fixed cost for opening:
  **binary variables.**
Limited capacity.
Biofuel supply chain design (Bai, Ouyang, Pang)

Possible locations for fuel processing.
Fixed cost for opening:
  **binary variables**.
Limited capacity.

Farm locations.
Biofuel supply chain design (Bai, Ouyang, Pang)

Possible locations for fuel processing.
Fixed cost for opening: **binary variables**.
Limited capacity.

Farm locations.

Biofuel company sets price for corn at each open location.
Total amount of corn available for biofuel is limited.
Farmers compete to supply corn: **complementary variables**.
Biofuel supply chain design (Bai, Ouyang, Pang)

Possible locations for fuel processing.
Fixed cost for opening: **binary variables**.
Limited capacity.

Farm locations.

Biofuel company sets price for corn at each open location.
Total amount of corn available for biofuel is limited.
Farmers compete to supply corn: **complementary variables**.
Bilevel programs

Bilevel program:

\[
\begin{align*}
\min_{x,y} \quad & f(x, y) \\
\text{s.t.} \quad & g(x, y) \leq 0 \\
& y \text{ binary} \\
& x \in \arg\min_x \{ h(x) : r(x) \leq v(y) \}
\end{align*}
\]
Bilevel programs

Bilevel program:

\[
\begin{align*}
\min_{x,y} & \quad f(x, y) \\
\text{s.t.} & \quad g(x, y) \leq 0 \\
& \quad y \text{ binary} \\
& \quad x \in \arg\min_x \{ h(x) : \quad r(x) \leq \nu(y) \}
\end{align*}
\]

DC-MPCC:

\[
\begin{align*}
\min_{x,y,z} & \quad f(x, y) \\
\text{s.t.} & \quad g(x, y) \leq 0 \\
& \quad y \text{ binary} \\
& \quad \nabla h(x) + \nabla r(x)z = 0 \\
& \quad 0 \leq \nu(y) - r(x) \perp z \geq 0
\end{align*}
\]
Bilevel programs

Bilevel program:
\[
\begin{align*}
\text{min}_{x,y} & \quad f(x, y) \\
\text{s.t.} & \quad g(x, y) \leq 0 \\
y & \text{binary} \\
x & \in \text{argmin}_x \{h(x) : r(x) \leq v(y)\}
\end{align*}
\]

DC-MPCC:
\[
\begin{align*}
\text{min}_{x,y,z} & \quad f(x, y) \\
\text{s.t.} & \quad g(x, y) \leq 0 \\
y & \text{binary} \\
\nabla h(x) + \nabla r(x)z & = 0 \\
0 & \leq v(y) - r(x) \perp z \geq 0
\end{align*}
\]

Equivalent if \(h(x)\) and \(r(x)\) are convex, and a constraint qualification holds for the subproblems.
Our Standard Form Optimization Problem

\[
\begin{align*}
\min_{x, y, w} & \quad g^T x + x^T Q x \\
\text{s.t.} & \quad A_I x + B_I y + C_I w \geq b_I \\
& \quad A_E x + B_E y + C_E w = b_E \\
& \quad 0 \leq y \perp w \geq 0
\end{align*}
\]

(QPCC)

\[y, w \in \mathbb{R}^{nc} \text{ and } Q \in \mathbb{R}^{nx \times nx} \text{ is a positive semi-definite matrix.}\]
Our Standard Form Optimization Problem

\[
\begin{align*}
\min_{x,y,w} & \quad g^T x + x^T Q x \\
\text{s.t.} & \quad A_l x + B_l y + C_l w \geq b_l \\
& \quad A_E x + B_E y + C_E w = b_E \\
& \quad 0 \leq y \perp w \geq 0
\end{align*}
\]

(QPCC)

- \( y, w \in \mathbb{R}^{n_c} \) and \( Q \in \mathbb{R}^{n_x \times n_x} \) is a positive semi-definite matrix.

- Non-convex, disjunctive problem.
Our Standard Form Optimization Problem

\[
\begin{align*}
\min_{x,y,w} & \quad g^T x + x^T Q x \\
\text{s.t.} & \quad A_I x + B_I y + C_I w \geq b_I \\
& \quad A_{EI} x + B_{EI} y + C_{EI} w = b_E \\
& \quad 0 \leq y \perp w \geq 0
\end{align*}
\] (QPCC)

- \( y, w \in \mathbb{R}^{nc} \) and \( Q \in \mathbb{R}^{nx \times nx} \) is a positive semi-definite matrix.
- Non-convex, disjunctive problem.
- We will discuss the addition of binary variables later.
Related work

- Local solutions

  ▶ NLP formulations: e.g. Filter-MPEC (Fletcher et al., 2004), KNITRO (Byrd et al., 2006).
  ▶ Regularization methods (Sholtes, 2000; Ralph & Wright, 2004).

- Global solutions

  ▶ Logical Benders Decomposition (LBD) for obtaining global solutions
  ▶ Linear programs with complementarity constraints (Hu, Mitchell, Pang, Bennett, Kunapuli 2008).
  ▶ Quadratic programs with complementarity constraints (Bai, Mitchell & Pang, 2013).
  ▶ Explores complementarity pieces.
  ▶ Discards pieces by generating cuts.
Related work

- Local solutions
  - NLP formulations: e.g. Filter-MPEC (Fletcher et al., 2004), KNITRO (Byrd et al., 2006).
  - Regularization methods (Sholtes, 2000; Ralph & Wright, 2004).

- Global solutions
    - Linear programs with complementarity constraints (Hu, Mitchell, Pang, Bennett, Kunapuli 2008).
    - Quadratic programs with complementarity constraints (Bai, Mitchell & Pang, 2013).
      - Explores complementarity pieces.
      - Discards pieces by generating cuts.
Related work

- Local solutions
  - NLP formulations: e.g. Filter-MPEC (Fletcher et al., 2004), KNITRO (Byrd et al., 2006).
  - Regularization methods (Sholtes, 2000; Ralph & Wright, 2004).
Related work

- Local solutions
  - NLP formulations: e.g. Filter-MPEC (Fletcher et al., 2004), KNITRO (Byrd et al., 2006).
  - Regularization methods (Sholtes, 2000; Ralph & Wright, 2004).

- Global solutions
Related work

- **Local solutions**
  - NLP formulations: e.g. Filter-MPEC (Fletcher et al., 2004), KNITRO (Byrd et al., 2006).
  - Regularization methods (Sholtes, 2000; Ralph & Wright, 2004).

- **Global solutions**
Related work

- **Local solutions**
  - NLP formulations: e.g. Filter-MPEC (Fletcher et al., 2004), KNITRO (Byrd et al., 2006).
  - Regularization methods (Sholtes, 2000; Ralph & Wright, 2004).

- **Global solutions**

- **Logical Benders Decomposition (LBD) for obtaining global solutions**
Related work

- **Local solutions**
  - NLP formulations: e.g. Filter-MPEC (Fletcher et al., 2004), KNITRO (Byrd et al., 2006).
  - Regularization methods (Sholtes, 2000; Ralph & Wright, 2004).

- **Global solutions**

- **Logical Benders Decomposition (LBD) for obtaining global solutions**
  - Linear programs with complementarity constraints (Hu, Mitchell, Pang, Bennett, Kunapuli 2008).
Related work

- **Local solutions**
  - NLP formulations: e.g. Filter-MPEC (Fletcher et al., 2004), KNITRO (Byrd et al., 2006).
  - Regularization methods (Sholties, 2000; Ralph & Wright, 2004).

- **Global solutions**

- **Logical Benders Decomposition (LBD) for obtaining global solutions**
  - Linear programs with complementarity constraints (Hu, Mitchell, Pang, Bennett, Kunapuli 2008).
  - Quadratic programs with complementarity constraints (Bai, Mitchell & Pang, 2013).
Related work

- **Local solutions**
  - NLP formulations: e.g. Filter-MPEC (Fletcher et al., 2004), KNITRO (Byrd et al., 2006).
  - Regularization methods (Sholtes, 2000; Ralph & Wright, 2004).

- **Global solutions**

- **Logical Benders Decomposition (LBD) for obtaining global solutions**
  - Linear programs with complementarity constraints (Hu, Mitchell, Pang, Bennett, Kunapuli 2008).
  - Quadratic programs with complementarity constraints (Bai, Mitchell & Pang, 2013).
  - Explores complementarity pieces.
Related work

- **Local solutions**
  - NLP formulations: e.g. Filter-MPEC (Fletcher et al., 2004), KNITRO (Byrd et al., 2006).
  - Regularization methods (Sholtes, 2000; Ralph & Wright, 2004).

- **Global solutions**

- **Logical Benders Decomposition (LBD) for obtaining global solutions**
  - Linear programs with complementarity constraints (Hu, Mitchell, Pang, Bennett, Kunapuli 2008).
  - Quadratic programs with complementarity constraints (Bai, Mitchell & Pang, 2013).
  - Explores complementarity pieces.
  - Discards pieces by generating cuts.
Primal Piece

Given a binary variable $p \in \{0, 1\}^{nc}$, the $p$-piece **primal** problem is...
Primal Piece

Given a binary variable \( p \in \{0, 1\}^{nc} \), the \( p \)-piece primal problem is

\[
\phi_P(p) \triangleq \min_{x, y, w} \quad g^T x + \frac{1}{2} x^T Q x \\
\text{s.t.} \quad A_I x + B_I y + C_I w & \leq b_I \quad (\mu_I) \\
A_E x + B_E y + C_E w & = b_E \quad (\mu_E) \\
w_i & \leq 0, \ i : p_i = 0 \quad (\lambda^w_i) \\
y_i & \leq 0, \ i : p_i = 1 \quad (\lambda^y_i) \\
w, y & \geq 0.
\]

Therefore, (QPCC) is equivalent to solving min \( p \in \{0, 1\}^{nc} \) \( \phi_P(p) \).
Primal Piece

Given a binary variable \( p \in \{0, 1\}^{nc} \), the \( p \)-piece primal problem is

\[
\phi_P(p) \triangleq \min_{x; y; w} \quad g^T x + \frac{1}{2} x^T Q x
\]

s.t. \[
A I x + B I y + C I w \leq b I \quad (\mu I) \\
A_E x + B_E y + C_E w = b_E \quad (\mu_E) \\
w_i \leq 0, \ i: p_i = 0 \quad (\lambda_i^w) \\
y_i \leq 0, \ i: p_i = 1 \quad (\lambda_i^y) \\
w, y \geq 0.
\]

Therefore, (QPCC) is equivalent to solving \( \min_{p \in \{0,1\}^{nc}} \phi_P(p) \).
Outline

Logical Benders Decomposition on QPCC - Simplified Algorithm

- **MP**:\( MP = \emptyset \)
- **Dual unbounded**
- **Dual**\( = 0 \)
- **Primal**
- **Add Cut**
- **Update U**
- **incumbent**
- **Finite**
- **Infeasible**
- **Unbounded**

\( p \)
Complementarity pieces $p$ are selected in a master problem.
Logical Benders Decomposition on QPCC - Simplified Algorithm

If the dual piece is unbounded, add cut to discard piece by infeasibility. Cuts have the form

\[ \sum_{i \in C^w} p_i + \sum_{i \in C^y} (1 - p_i) \geq 1. \]
**Outline**

**Logical Benders Decomposition on QPCC - Simplified Algorithm**

If primal is finite, update incumbent and add cut to discard piece. Cuts also have the form

\[
\sum_{i \in C^w} p_i + \sum_{i \in C^y} (1 - p_i) \geq 1.
\]
Outline

Logical Benders Decomposition on QPCC - Simplified Algorithm

If primal is unbounded, QPCC is unbounded. STOP!
Logical Benders Decomposition on QPCC - Simplified Algorithm

Method continues until master problem is infeasible.
Outline

Logical Benders Decomposition on QPCC - Simplified Algorithm

This talk:
- Piece selection (MP)
- Cut strengthening (Add Cut)
Outline

1 Introduction
   - QPCC
   - Simplified Method

2 Master Problem
   - How to select $p$?

3 Cut Strengthening
   - $\ell_1$-norm sparsification
   - Tree guided sparsification
   - Hybrid Sparsification

4 Numerical Results

5 Extensions
   - Binary Variables

6 Summary & Future Research
How to select $p$?

- Suppose we are solving QPCC via a **Branch-and-Bound** approach.
How to select $p$?

- Suppose we are solving QPCC via a **Branch-and-Bound** approach.
How to select $p$?

Suppose we are solving QPCC via a **Branch-and-Bound** approach.

\[ w_1 = 0 \]
How to select $p$?

- Suppose we are solving QPCC via a **Branch-and-Bound** approach.
How to select $p$?

- Suppose we are solving QPCC via a **Branch-and-Bound** approach.
Suppose we are solving QPCC via a **Branch-and-Bound** approach.
How to select $p$?

- Suppose we are solving QPCC via a **Branch-and-Bound** approach.
How to select $p$?

- Suppose we are solving QPCC via a **Branch-and-Bound** approach.
- Two nodes have been fathomed.
How to select \( p \)?

- Suppose we are solving QPCC via a **Branch-and-Bound** approach.
- Two nodes have been fathomed.
- The fathomed nodes may be interpreted as cuts.
  
  \[
  \text{(Recall: } p_i = 0 \leftrightarrow w_i = 0, \ p_i = 1 \leftrightarrow y_i = 0) \\
  \begin{align*}
  & \quad \check{p_1} + p_2 + p_3 \geq 1 \\
  & \quad (1 - p_1) + p_3 \geq 1
  \end{align*}
  \]

\[\begin{array}{c}
1 \\
2 \\
3 \\
4 \\
5 \\
6 \\
8
\end{array}\]

\[\begin{array}{c}
w_1 = 0 \\
w_3 = 0 \\
y_1 = 0 \\
w_3 = 0 \\
w_2 = 0
\end{array}\]
How to select \( p \)?

- Suppose we are solving QPCC via a **Branch-and-Bound** approach.
- Two nodes have been fathomed.
- The fathomed nodes may be interpreted as cuts.

\[
\text{(Recall: } p_i = 0 \leftrightarrow w_i = 0, \ p_i = 1 \leftrightarrow y_i = 0)\]

- \( p_1 + p_2 + p_3 \geq 1 \)
- \( (1 - p_1) + p_3 \geq 1 \)
Visualizing the cuts in the Master Problem
Visualizing the cuts in the Master Problem
Visualizing the cuts in the Master Problem

\[ p = (0, 0, 0) \]
Visualizing the cuts in the Master Problem

\[ p = (0, 0, 0) \]
\[ p_1 + p_2 + p_3 \geq 1 \]
Visualizing the cuts in the Master Problem

\[ p = (0, 0, 0) \]
\[ p_1 + p_2 + p_3 \geq 1 \]
\[ p = (1, \#, 0) \]
Visualizing the cuts in the Master Problem

\[ p = (0, 0, 0) \]
\[ p_1 + p_2 + p_3 \geq 1 \]

\[ p = (1, \#, 0) \]
\[ (1 - p_1) + p_3 \geq 1 \]

5 points feasible in Master Problem
Visualizing the cuts in the Master Problem

\[ p = (0, 0, 0) \]
\[ p_1 + p_2 + p_3 \geq 1 \]

\[ p = (1, \#, 0) \]
\[ (1 - p_1) + p_3 \geq 1 \]

5 points feasible in Master Problem

Try to **diversify** search
Outline of Master Problem heuristic

Idea: “Construct a tree” from a set of cuts and pick a leaf $p$ from a branch to increase likelihood of being fathomed close to the root. We are trying to diversify the search.
Outline of Master Problem heuristic

**Input:** A set of cuts \( \{C_k^w, C_k^y\} \).

**Output:** A leaf \( \hat{\rho} \), a path \( P \).

- \( p_1 + p_2 + p_3 \geq 1 \)
- \( (1 - p_1) + p_3 \geq 1 \)

\[
\begin{align*}
p_i = 0 &\iff w_i = 0 \\
p_i = 1 &\iff y_i = 0
\end{align*}
\]
Outline of Master Problem heuristic

**Input:** A set of cuts $\{C_k^w, C_k^y\}$.

**Output:** A leaf $p$, a path $P$.

- $p_1 + p_2 + p_3 \geq 1$
- $(1 - p_1) + p_3 \geq 1$

Branch on most explored component $\hat{j}$.

$$\hat{j} = \arg \max_j |\{k : j \in C_k^w \cup C_k^y\}|$$

- $p_i = 0 \iff w_i = 0$
- $p_i = 1 \iff y_i = 0$
Outline of Master Problem heuristic

**Input:** A set of cuts \( \{ C^w_k, C^y_k \} \).

**Output:** A leaf \( \hat{p} \), a path \( P \).

- Branch on most explored component \( \hat{j} \).
- Choose least explored side of branch.
  \[ \hat{p}_j = 0 \text{ if } |\{ k : \hat{j} \in C^w_k \}| \leq |\{ k : \hat{j} \in C^y_k \}|; \hat{p}_j = 1 \text{ otherwise.} \]

\[ \begin{align*}
p_1 + p_2 + p_3 & \geq 1 \\
(1 - p_1) + p_3 & \geq 1
\end{align*} \]

\[ p_i = 0 \iff w_i = 0 \]

\[ p_i = 1 \iff y_i = 0 \]
Outline of Master Problem heuristic

**Input:** A set of cuts \( \{ C_k^w, C_k^y \} \).

**Output:** A leaf \( \hat{p} \), a path \( P \).

- \( p_1 + p_2 + p_3 \geq 1 \)
- \( (1 - p_1) + p_3 \geq 1 \)

- Branch on most explored component \( \hat{j} \).
- Choose least explored side of branch.
- Add \( \hat{j} \) to \( P \). Remove \( \hat{j} \) from branching candidates.

\[
\begin{align*}
p_i = 0 & \iff w_i = 0 \\
p_i = 1 & \iff y_i = 0
\end{align*}
\]
Outline of Master Problem heuristic

**Input:** A set of cuts \( \{ C_k^w, C_k^y \} \).

**Output:** A leaf \( \hat{p} \), a path \( P \).

- \( p_1 + p_2 + p_3 \geq 1 \)
- \( (1 - p_1) + p_3 \geq 1 \)

Branch on **most explored component** \( \hat{j} \).

Choose **least explored side** of branch.

Add \( \hat{j} \) to \( P \). Remove \( \hat{j} \) from branching candidates.

\[
\begin{align*}
p_1 &= 0 \iff w_i = 0 \\
p_1 &= 1 \iff y_i = 0
\end{align*}
\]
Outline of Master Problem heuristic

Input: A set of cuts \( \{ C_k^w, C_k^y \} \).

Output: A leaf \( \hat{p} \), a path \( P \).

- Branch on most explored component \( \hat{j} \).
- Choose least explored side of branch.
- Add \( \hat{j} \) to \( P \). Remove \( \hat{j} \) from branching candidates.

\[
\begin{align*}
p_1 + p_2 + p_3 & \geq 1 \\
(1 - p_1) + p_3 & \geq 1
\end{align*}
\]

\[
p_i = 0 \iff w_i = 0 \\
p_i = 1 \iff y_i = 0
\]
Outline of Master Problem heuristic

**Input:** A set of cuts $\{C_k^w, C_k^y\}$.

**Output:** A leaf $\hat{p}$, a path $P$.

- $p_1 + p_2 + p_3 \geq 1$
- $(1 - p_1) + p_3 \geq 1$

- Branch on most explored component $\hat{j}$.
- Choose least explored side of branch.
- Add $\hat{j}$ to $P$. Remove $\hat{j}$ from branching candidates.

$(w_1 = 0, y_1 = 0, 0)$
$(w_2 = 0, y_2 = 0, 0)$
$(w_3 = 0, y_3 = 0, 0)$
$(w_1 = 0, y_1 = 0, 0)$
$(w_2 = 0, y_2 = 0, 0)$

$p_i = 0 \iff w_i = 0$
$p_i = 1 \iff y_i = 0$
Resolution
We have the two cuts:

\[ p_1 + p_2 + p_3 \geq 1, \quad (1 - p_1) + p_3 \geq 1 \]

Resolution is a procedure for generating satisfiability cuts that improve the LP relaxation of the Master Problem. Here, we can get the additional cut

\[ p_2 + p_3 \geq 1. \]

In this example, running our heuristic gives the same leaf \( \hat{p} \), albeit with a different branching order:
Resolution

We have the two cuts:

\[ p_1 + p_2 + p_3 \geq 1, \quad (1 - p_1) + p_3 \geq 1 \]

Resolution is a procedure for generating satisfiability cuts that improve the LP relaxation of the Master Problem. Here, we can get the additional cut

\[ p_2 + p_3 \geq 1. \]

In this example, running our heuristic gives the same leaf \( \hat{p} \), albeit with a different branching order:

\[ p_i = 0 \iff w_i = 0 \]
\[ p_i = 1 \iff y_i = 0 \]
Resolution
We have the two cuts:

\[ p_1 + p_2 + p_3 \geq 1, \quad (1 - p_1) + p_3 \geq 1 \]

Resolution is a procedure for generating satisfiability cuts that improve the LP relaxation of the Master Problem. Here, we can get the additional cut

\[ p_2 + p_3 \geq 1. \]

In this example, running our heuristic gives the same leaf \( \hat{p} \), albeit with a different branching order:
Resolution
We have the two cuts:

\[ p_1 + p_2 + p_3 \geq 1, \quad (1 - p_1) + p_3 \geq 1 \]

Resolution is a procedure for generating satisfiability cuts that improve the LP relaxation of the Master Problem. Here, we can get the additional cut

\[ p_2 + p_3 \geq 1. \]

In this example, running our heuristic gives the same leaf \( \hat{p} \), albeit with a different branching order:
Resolution
We have the two cuts:

\[ p_1 + p_2 + p_3 \geq 1, \quad (1 - p_1) + p_3 \geq 1 \]

Resolution is a procedure for generating satisfiability cuts that improve the LP relaxation of the Master Problem. Here, we can get the additional cut

\[ p_2 + p_3 \geq 1. \]

In this example, running our heuristic gives the same leaf \( \hat{p} \), albeit with a different branching order:

\[ w_3 = 0 \quad y_3 = 0 \]
\[ w_2 = 0 \quad y_2 = 0 \]
Resolution
We have the two cuts:

\[ p_1 + p_2 + p_3 \geq 1, \quad (1 - p_1) + p_3 \geq 1 \]

Resolution is a procedure for generating satisfiability cuts that improve the LP relaxation of the Master Problem. Here, we can get the additional cut

\[ p_2 + p_3 \geq 1. \]

In this example, running our heuristic gives the same leaf \( \hat{p} \), albeit with a different branching order:
**Resolution**

We have the two cuts:

\[ p_1 + p_2 + p_3 \geq 1, \quad (1 - p_1) + p_3 \geq 1 \]

**Resolution** is a procedure for generating satisfiability cuts that improve the LP relaxation of the Master Problem. Here, we can get the additional cut

\[ p_2 + p_3 \geq 1. \]

In this example, running our heuristic gives the same leaf \( \hat{p} \), albeit with a different branching order:
Outline

1. Introduction
   - QPCC
   - Simplified Method

2. Master Problem
   - How to select $p$?

3. Cut Strengthening
   - $\ell_1$-norm sparsification
   - Tree guided sparsification
   - Hybrid Sparsification

4. Numerical Results

5. Extensions
   - Binary Variables

6. Summary & Future Research
Cut Strengthening

- Cuts have the form

$$\sum_{i \in C^w} p_i + \sum_{i \in C^y} (1 - p_i) \geq 1.$$
Cut Strengthening

- Cuts have the form
  \[ \sum_{i \in C^w} p_i + \sum_{i \in C^y} (1 - p_i) \geq 1. \]

- They correspond to primal relaxation
  \[
  \phi_P(C, p) = \min_{x, y, w} \quad g^T x + x^T Qx \\
  \text{s.t.} \quad A_I x + B_I y + C_I w \leq b_I, \\
  \quad A_E x + B_E y + C_E w = b_E, \\
  \quad w_i \leq 0 \quad i \in C^w \\
  \quad y_i \leq 0 \quad i \in C^y \\
  \quad w, y \geq 0.
  \]
Cut Strengthening

- Cuts have the form
  \[ \sum_{i \in C^w} p_i + \sum_{i \in C^y} (1 - p_i) \geq 1. \]

- They correspond to primal relaxation
  \[ \phi_P(C, p) = \min_{x, y, w} g^T x + x^T Qx \]
  \[ \text{s.t.} \quad A_I x + B_I y + C_I w \leq b_I, \]
  \[ A_E x + B_E y + C_E w = b_E, \]
  \[ w_i \leq 0 \quad i \in C^w \]
  \[ y_i \leq 0 \quad i \in C^y \]
  \[ w, y \geq 0. \]

- Would like to find smallest sets \( C^w \) and \( C^y \) such that \( \phi_P(C, p) \geq U \).
Weighted $\ell_1$-norm sparsification (WL1)

- Formulate an $\ell_0$-norm dual problem

$$\phi_{Ds}(p) = \min_{\mu_I, \mu_E, \lambda^w, \lambda^y} \|\lambda^w\|_0 + \|\lambda^y\|_0$$

subject to:

$$-A_i^T \mu_I + A_E^T \mu_E = 0$$
$$-B_i^T \mu_I + B_E^T \mu_E - \lambda^y \leq 0$$
$$-C_i^T \mu_I + C_E^T \mu_E - \lambda^w \leq 0$$

$$p^T \lambda^w + (1-p)^T \lambda^y = 0$$
$$-b_i^T \mu_I + b_E^T \mu_E = 1$$

$$\mu_I, \lambda^w, \lambda^y \geq 0.$$ 

- Look for sparse $\lambda$, since $C^w = \{i : \lambda_i^w > 0\}$, $C^y = \{i : \lambda_i^y > 0\}$ in the cut

$$\sum_{i \in C^w} p_i + \sum_{i \in C^y} (1 - p_i) \geq 1.$$
Weighted $\ell_1$-norm sparsification (WL1)

- Formulate an $\ell_1$-norm dual problem

$$\phi_{D_s}(p) = \min_{\mu_I, \mu_E, \lambda^w, \lambda^y} \|\lambda^w\|_1 + \|\lambda^y\|_1$$

subject to

$$-A_i^T \mu_I + A_E^T \mu_E = 0$$

$$-B_i^T \mu_I + B_E^T \mu_E - \lambda^y \leq 0$$

$$-C_i^T \mu_I + C_E^T \mu_E - \lambda^w \leq 0$$

$$p^T \lambda^w + (1 - p)^T \lambda^y = 0$$

$$-b_i^T \mu_I + b_E^T \mu_E = 1$$

$$\mu_I, \lambda^w, \lambda^y \geq 0.$$ 

- Look for sparse $\lambda$, since $C^w = \{i : \lambda^w_i > 0\}$, $C^y = \{i : \lambda^y_i > 0\}$ in the cut

$$\sum_{i \in C^w} p_i + \sum_{i \in C^y} (1 - p_i) \geq 1.$$
Weighted $\ell_1$-norm sparsification (WL1)

- Formulate an $\ell_1$-norm dual problem

$$\phi_{D_s}(p) = \min_{\mu_I, \mu_E, \lambda^w, \lambda^y} \sum_i (\omega_i^k \lambda_i^w + \gamma_i^k \lambda_i^y)$$

subject to

$$-A_i^T \mu_I + A_E^T \mu_E = 0$$
$$-B_i^T \mu_I + B_E^T \mu_E - \lambda^y \leq 0$$
$$-C_i^T \mu_I + C_E^T \mu_E - \lambda^w \leq 0$$

$$p^T \lambda^w + (1 - p)^T \lambda^y = 0$$

$$-b_i^T \mu_I + b_E^T \mu_E = 1$$

$$\mu_I, \lambda^w, \lambda^y \geq 0.$$

Remarks

- To solve $\ell_1$ we use a weighted iterative procedure.

$$\omega_i^{k+1} = \frac{1}{\max\{\lambda_{i,k+1}^w, \varepsilon\}} \quad \text{and} \quad \gamma_i^{k+1} = \frac{1}{\max\{\lambda_{i,k+1}^y, \varepsilon\}}$$
Tree guided sparsification (Tree)
We have piece $p$ and path $P$. 
Tree guided sparsification (Tree)

We have piece $p$ and path $P$.

**Idea:** Relax complementarities of primal piece $p$, one by one, following the path $P$. 
Tree guided sparsification (Tree)

We have piece $p$ and path $P$.

**Idea:** Relax complementarities of primal piece $p$, one by one, following the path $P$.

- From the example we had $p = (0, 1, 1)$ and $P = 3 \rightarrow 1 \rightarrow 2$ (root to leaf).
Tree guided sparsification (Tree)
We have piece $p$ and path $P$.

**Idea:** Relax complementarities of primal piece $p$, one by one, following the path $P$.

- From the example we had $p = (0, 1, 1)$ and $P = 3 \rightarrow 1 \rightarrow 2$ (root to leaf).
- The corresponding cut is $C^w := \{1\}$ and $C^y := \{2, 3\}.$
Tree guided sparsification (Tree)

We have piece \( p \) and path \( P \).

**Idea:** Relax complementarities of primal piece \( p \), one by one, following the path \( P \).

- From the example we had \( p = (0, 1, 1) \) and \( P = 3 \rightarrow 1 \rightarrow 2 \) (root to leaf).
- The corresponding cut is \( C^w := \{1\} \) and \( C^y := \{2, 3\} \).
  - Relax last element of \( P \) (or, remove 2 from \( C^y \)).
Tree guided sparsification (Tree)
We have piece $p$ and path $P$.

Idea: Relax complementarities of primal piece $p$, one by one, following the path $P$.

- From the example we had $p = (0, 1, 1)$ and $P = 3 \rightarrow 1 \rightarrow 2$ (root to leaf).
- The corresponding cut is $C^w := \{1\}$ and $C^y := \{2, 3\}$.
  - Relax last element of $P$ (or, remove 2 from $C^y$).
  - If the relaxation is larger or equal than $U$, we accept it.
Tree guided sparsification (Tree)

We have piece $p$ and path $P$.

**Idea:** Relax complementarities of primal piece $p$, one by one, following the path $P$.

- From the example we had $p = (0, 1, 1)$ and $P = 3 \rightarrow 1 \rightarrow 2$ (root to leaf).
- The corresponding cut is $C_w := \{1\}$ and $C_y := \{2, 3\}$.
  - Relax last element of $P$ (or, remove 2 from $C_y$).
  - If the relaxation is larger or equal than $U$, we accept it.
  - Otherwise, we fix that complementarity back (or, return 2 to $C_y$).
Tree guided sparsification (Tree)
We have piece $p$ and path $P$.

**Idea:** Relax complementarities of primal piece $p$, one by one, following the path $P$.

- From the example we had $p = (0, 1, 1)$ and $P = 3 \to 1 \to 2$ (root to leaf).
- The corresponding cut is $C^w := \{1\}$ and $C^y := \{2, 3\}$.
  - Relax last element of $P$ (or, remove 2 from $C^y$).
  - If the relaxation is larger or equal than $U$, we accept it.
  - Otherwise, we fix that complementarity back (or, return 2 to $C^y$).
  - Move onto next element of $P$. 
Tree guided sparsification (Tree)
We have piece $p$ and path $P$.

**Idea:** Relax complementarities of primal piece $p$, one by one, following the path $P$.

- From the example we had $p = (0, 1, 1)$ and $P = 3 \rightarrow 1 \rightarrow 2$ (root to leaf).
- The corresponding cut is $C^w := \{1\}$ and $C^y := \{2, 3\}$.
  - Relax last element of $P$ (or, remove 2 from $C^y$).
  - If the relaxation is larger or equal than $U$, we accept it.
  - Otherwise, we fix that complementarity back (or, return 2 to $C^y$).
  - Move onto next element of $P$.

Resulting cut is less likely to contain complementarities that are closer to the leaf $p$. 
Hybrid Sparsification (Hyb)

- Advantages & disadvantages

- **Advantage:** Solve few LPs ($\sim 6$ W1 iterations for $n_c = 100$).
- **Disadvantage:** Cut may not be minimal.

- **Advantage:** Cut cannot be sparsified further.
- **Disadvantage:** Could require many ($n_c$) QP solves.
Hybrid Sparsification (Hyb)

- Advantages & disadvantages
  - $\ell_1$-norm sparsification

**Advantage:** Solve few LPs ($\sim 6$ WL1 iterations for $n_c = 100$).

**Disadvantage:** Cut may not be minimal.

**Tree sparsification**

- **Advantage:** Cut cannot be sparsified further.
- **Disadvantage:** Could require many ($n_c$) QP solves.

Hybrid Sparsification

- First solve $\ell_1$-norm sparsification. Obtain $C_w$ and $C_y$.
- Then apply tree guided sparsification only over remaining elements of $C$. 
Hybrid Sparsification (Hyb)

- Advantages & disadvantages
  - $\ell_1$-norm sparsification
    - Advantage: Solve few LPs ($\sim 6$ WL1 iterations for $n_c = 100$).
  - Tree sparsification
    - Advantage: Cut cannot be sparsified further.
    - Disadvantage: Could require many ($n_c$) QP solves.
Advantages & disadvantages

- $\ell_1$-norm sparsification
  - Advantage: Solve few LPs ($\sim 6$ WL1 iterations for $n_c = 100$).
  - Disadvantage: Cut may not be minimal.

Hybrid Sparsification (Hyb)
Hybrid Sparsification (Hyb)

Advantages & disadvantages

- $\ell_1$-norm sparsification
  - Advantage: Solve few LPs ($\sim 6$ WL1 iterations for $n_c = 100$).
  - Disadvantage: Cut may not be minimal.

- Tree sparsification
Hybrid Sparsification (Hyb)

- Advantages & disadvantages
  - $\ell_1$-norm sparsification
    - Advantage: Solve few LPs ($\sim 6$ WL1 iterations for $n_c = 100$).
    - Disadvantage: Cut may not be minimal.
  - Tree sparsification
    - Advantage: Cut cannot be sparsified further.
Hybrid Sparsification (Hyb)

Advantages & disadvantages

- $\ell_1$-norm sparsification
  - Advantage: Solve few LPs ($\sim 6$ WL1 iterations for $n_c = 100$).
  - Disadvantage: Cut may not be minimal.

- Tree sparsification
  - Advantage: Cut cannot be sparsified further.
  - Disadvantage: Could require many ($n_c$) QP solves.
Hybrid Sparsification (Hyb)

- Advantages & disadvantages
  - \( \ell_1 \)-norm sparsification
    - Advantage: Solve few LPs (\( \sim 6 \) WL1 iterations for \( n_c = 100 \)).
    - Disadvantage: Cut may not be minimal.
  - Tree sparsification
    - Advantage: Cut cannot be sparsified further.
    - Disadvantage: Could require many \( n_c \) QP solves.

- Hybrid Sparsification

  First solve \( \ell_1 \)-norm sparsification. Obtain \( C_w \) and \( C_y \).
  Then apply tree guided sparsification only over remaining elements of \( C \).
Hybrid Sparsification (Hyb)

- **Advantages & disadvantages**
  - $\ell_1$-norm sparsification
    - Advantage: Solve few LPs ($\sim 6$ WL1 iterations for $n_c = 100$).
    - Disadvantage: Cut may not be minimal.
  - Tree sparsification
    - Advantage: Cut cannot be sparsified further.
    - Disadvantage: Could require many ($n_c$) QP solves.

- **Hybrid Sparsification**
  - First solve $\ell_1$-norm sparsification. Obtain $C^w$ and $C^y$. 
Hybrid Sparsification (Hyb)

Advantages & disadvantages

- \(\ell_1\)-norm sparsification
  - Advantage: Solve few LPs (\(\sim 6\) WL1 iterations for \(n_c = 100\)).
  - Disadvantage: Cut may not be minimal.

- Tree sparsification
  - Advantage: Cut cannot be sparsified further.
  - Disadvantage: Could require many \((n_c)\) QP solves.

Hybrid Sparsification

- First solve \(\ell_1\)-norm sparsification. Obtain \(C^w\) and \(C^y\).

- Then apply tree guided sparsification only over remaining elements of \(C\).
Split the problem

Upper bound helps getting tighter cuts.
Would like good starting upper bound $U$. 
Split the problem

Upper bound helps getting tighter cuts. Would like good starting upper bound $U$.
Pick large constant $T$ and solve two QPCCs:

Add constraints $w \leq Tp$, $y \leq T(1-p)$, $p$ binary, and $y_i + w_i \leq T$ for all $i$. Optimal value gives upper bound $U$, with a corresponding piece.
Split the problem

Upper bound helps getting tighter cuts.
Would like good starting upper bound $U$.
Pick large constant $T$ and solve two QPCCs:

$$y_i \leq T,$$
$$w_i \leq Tp,$$
$$p \text{ binary},$$
$$y_i + w_i \leq T$$ for all $i$.

Optimal value gives upper bound $U$, with a corresponding piece.
Split the problem

Upper bound helps getting tighter cuts. Would like good starting upper bound $U$. Pick large constant $T$ and solve two QPCCs:

- **Bounded feasible region, solve as MIQP.**
  
  Add constraints $w \leq Tp$, $y \leq T(1 - p)$, $p$ binary, and $y_i + w_i \leq T$ for all $i$.

  Optimal value gives **upper bound** $U$, with a corresponding piece.
Split the problem

Upper bound helps getting tighter cuts.
Would like good starting upper bound $U$. 
Pick large constant $T$ and solve two QPCCs:

- **Bounded feasible region, solve as MIQP.**
  
  Add constraints $w \leq Tp$, $y \leq T(1 - p)$,
  $p$ binary, and $y_i + w_i \leq T$ for all $i$.
  
  Optimal value gives **upper bound** $U$, with a corresponding piece.
Split the problem

Upper bound helps getting tighter cuts.
Would like good starting upper bound $U$.
Pick large constant $T$ and solve two QPCCs:

- Bounded feasible region, solve as MIQP.
  Add constraints $w \leq Tp$, $y \leq T(1 - p)$, $p$ binary, and $y_i + w_i \leq T$ for all $i$.

Optimal value gives **upper bound** $U$, with a corresponding piece.
Split the problem

Upper bound helps getting tighter cuts.
Would like good starting upper bound $U$.
Pick large constant $T$ and solve two QPCCs:

1. **Bounded feasible region, solve as MIQP.**
   Add constraints $w \leq Tp$, $y \leq T(1 - p)$,
   $p$ binary, and $y_i + w_i \leq T$ for all $i$.
   Optimal value gives **upper bound** $U$,
   with a corresponding piece.

2. **A region bounded away from the origin.**
   Add constraint $1^T y + 1^T w \geq T$.
   Solve using **logical Benders** approach.
   Can function as a “certificate of optimality”.

When use dual LPs to look for ray cuts,
    can add **linearizations** of the objective to the primal constraints.
Outline

1. Introduction
   - QPCC
   - Simplified Method

2. Master Problem
   - How to select $p$?

3. Cut Strengthening
   - $\ell_1$-norm sparsification
   - Tree guided sparsification
   - Hybrid Sparsification

4. Numerical Results

5. Extensions
   - Binary Variables

6. Summary & Future Research
## Numerical Results

**LPCC:**

<table>
<thead>
<tr>
<th>$n_c$</th>
<th>LBD (2008)</th>
<th>WL1</th>
<th>Hyb</th>
<th>Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>386.57</td>
<td>11.32</td>
<td>13.65</td>
<td>31.82</td>
</tr>
<tr>
<td></td>
<td>41.98</td>
<td>2.28</td>
<td>2.53</td>
<td>4.97</td>
</tr>
<tr>
<td></td>
<td>25.63</td>
<td>0.62</td>
<td>0.72</td>
<td>2.16</td>
</tr>
<tr>
<td></td>
<td>961.65</td>
<td>9.39</td>
<td>6.61</td>
<td>24.8</td>
</tr>
<tr>
<td></td>
<td>46.78</td>
<td>6.23</td>
<td>6.48</td>
<td>30.39</td>
</tr>
<tr>
<td></td>
<td>32.18</td>
<td>1.58</td>
<td>1.74</td>
<td>3.81</td>
</tr>
<tr>
<td></td>
<td>45.08</td>
<td>28.9</td>
<td>27.27</td>
<td>88.2</td>
</tr>
<tr>
<td></td>
<td>488.18</td>
<td>4.38</td>
<td>3.74</td>
<td>10.51</td>
</tr>
<tr>
<td></td>
<td>14.89</td>
<td>39.49</td>
<td>34.49</td>
<td>97.75</td>
</tr>
<tr>
<td></td>
<td>811.55</td>
<td>8.55</td>
<td>13.58</td>
<td>40.56</td>
</tr>
<tr>
<td>300</td>
<td>190.92</td>
<td>132.69</td>
<td>241.58</td>
<td>637.43</td>
</tr>
<tr>
<td>&gt; 1 hour</td>
<td>101.62</td>
<td>92.36</td>
<td>203.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>528.83</td>
<td>1936.81</td>
<td>1805.64</td>
<td>&gt; 1 hour</td>
</tr>
<tr>
<td>&gt; 1 hour</td>
<td>135.11</td>
<td>352.3</td>
<td>815.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>553.83</td>
<td>343.34</td>
<td>295.64</td>
<td>919.32</td>
</tr>
<tr>
<td>&gt; 1 hour</td>
<td>213.85</td>
<td>&gt; 1 hour</td>
<td>&gt; 1 hour</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>264.25</td>
<td>235.96</td>
<td>686.39</td>
</tr>
<tr>
<td></td>
<td>112.72</td>
<td>159.03</td>
<td>207.6</td>
<td>790.72</td>
</tr>
<tr>
<td></td>
<td>872.64</td>
<td>569.24</td>
<td>387.07</td>
<td>992.7</td>
</tr>
<tr>
<td>&gt; 1 hour</td>
<td>144.85</td>
<td>221.86</td>
<td>774.39</td>
<td></td>
</tr>
</tbody>
</table>

MIQP preprocessing, $T = 100$. 
### Numerical Results

#### LPCC:

<table>
<thead>
<tr>
<th>$n_c$</th>
<th>Total Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WL1</td>
</tr>
<tr>
<td>100</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>29</td>
<td>16</td>
</tr>
<tr>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>88</td>
<td>75</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
</tr>
<tr>
<td>120</td>
<td>91</td>
</tr>
<tr>
<td>25</td>
<td>34</td>
</tr>
<tr>
<td>100</td>
<td>171</td>
</tr>
<tr>
<td>300</td>
<td>1113</td>
</tr>
<tr>
<td>150</td>
<td>&gt; 3000</td>
</tr>
<tr>
<td>200</td>
<td>235</td>
</tr>
<tr>
<td>115</td>
<td>147</td>
</tr>
<tr>
<td>378</td>
<td>255</td>
</tr>
<tr>
<td>101</td>
<td>148</td>
</tr>
</tbody>
</table>

MIQP preprocessing, $T = 100$. 
## Numerical Results

### QPCC

<table>
<thead>
<tr>
<th>$n_c$</th>
<th>Total Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No outer box</td>
</tr>
<tr>
<td>SAT</td>
<td>Tree</td>
</tr>
<tr>
<td>50</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>15</td>
</tr>
<tr>
<td>38</td>
<td>22</td>
</tr>
<tr>
<td>34</td>
<td>17</td>
</tr>
<tr>
<td>30</td>
<td>19</td>
</tr>
<tr>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>37</td>
<td>17</td>
</tr>
<tr>
<td>$&gt;100$</td>
<td>37</td>
</tr>
<tr>
<td>$&gt;100$</td>
<td>15</td>
</tr>
<tr>
<td>38</td>
<td>23</td>
</tr>
<tr>
<td>$&gt;100$</td>
<td>8</td>
</tr>
<tr>
<td>100</td>
<td></td>
</tr>
<tr>
<td>137</td>
<td>35</td>
</tr>
<tr>
<td>-</td>
<td>62</td>
</tr>
<tr>
<td>102</td>
<td>23</td>
</tr>
<tr>
<td>66</td>
<td>37</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>-</td>
<td>16</td>
</tr>
<tr>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>-</td>
<td>34</td>
</tr>
<tr>
<td>47</td>
<td>3</td>
</tr>
<tr>
<td>-</td>
<td>37</td>
</tr>
</tbody>
</table>

MIQP preprocessing, Hybrid, $T = 1000$.  

John E. Mitchell (RPI)
Outline

1. Introduction
   - QPCC
   - Simplified Method

2. Master Problem
   - How to select $p$?

3. Cut Strengthening
   - $\ell_1$-norm sparsification
   - Tree guided sparsification
   - Hybrid Sparsification

4. Numerical Results

5. Extensions
   - Binary Variables

6. Summary & Future Research
Binary Variables

- The previous settings can be extended to a binary vector $z$ \((0 \leq z \perp 1 - z \geq 0)\).
Binary Variables

- The previous settings can be extended to a binary vector \( z \) \((0 \leq z \perp 1 - z \geq 0)\).
- Design variables are now a tuple \((\hat{p}^c, \hat{p}^b)\).
Binary Variables

- The previous settings can be extended to a binary vector $z$ ($0 \leq z \perp 1 - z \geq 0$).
- Design variables are now a tuple $(\hat{p}^c, \hat{p}^b)$.
- Cuts now have the form

$$\sum_{i \in C^0} p_i^b + \sum_{i \in C^1} (1 - p_i^b) + \sum_{i \in C^w} p_i^c + \sum_{i \in C^y} (1 - p_i^c) \geq 1$$
Binary Variables

- The previous settings can be extended to a binary vector $z$ ($0 \leq z \perp 1 - z \geq 0$).
- Design variables are now a tuple $(\hat{p}^c, \hat{p}^b)$.
- Cuts now have the form which can be strengthened to

$$\sum_{i \in C^0} \bar{g}_i p_i^b + \sum_{i \in C^1} \bar{h}_i (1 - p_i^b) + \sum_{i \in C^w} p_i^c + \sum_{i \in C^y} (1 - p_i^c) \geq 1$$

where $\bar{g}, \bar{h} \leq 1$. 
Binary Variables

- The previous settings can be extended to a binary vector \( z \) (\( 0 \leq z \perp 1 − z \geq 0 \)).
- Design variables are now a tuple \((\hat{p}^c, \hat{p}^b)\).
- Cuts now have the form which can be strengthened to

\[
\sum_{i \in C^0} \bar{g}_i p^b_i + \sum_{i \in C^1} \bar{h}_i (1 - p^b_i) + \sum_{i \in C^w} p^c_i + \sum_{i \in C^y} (1 - p^c_i) \geq 1
\]

where \( \bar{g}, \bar{h} \leq 1 \).

- Both \( \ell_1 \) and tree-guided sparsification remain the same over this extension.
Outline

1. Introduction
   - QPCC
   - Simplified Method

2. Master Problem
   - How to select \( p \)?

3. Cut Strengthening
   - \( \ell_1 \)-norm sparsification
   - Tree guided sparsification
   - Hybrid Sparsification

4. Numerical Results

5. Extensions
   - Binary Variables

6. Summary & Future Research
Summary & Future Research

- Summary

Future research

- Extend to larger problems.
- Improve bounds during tree sparsification.
- Extend tree guided master problem for binary variables.
Summary & Future Research

- **Summary**
  - Presented a global optimization method for QPCCs with finite termination based on logical Benders decomposition.

- **Future research**
  - Extend to larger problems.
  - Improve bounds during tree sparsification.
  - Extend tree guided master problem for binary variables.
Summary & Future Research

Summary

▶ Presented a global optimization method for QPCCs with finite termination based on logical Benders decomposition.

▶ Proposed a tree guided master problem heuristic to search for unexplored pieces.
Summary & Future Research

Summary

- Presented a global optimization method for QPCCs with finite termination based on logical Benders decomposition.
- Proposed a tree guided master problem heuristic to search for unexplored pieces.
- Proposed two methods for cut sparsification.

Future research

- Extend to larger problems.
- Improve bounds during tree sparsification.
- Extend tree guided master problem for binary variables.
Summary & Future Research

- **Summary**
  - Presented a global optimization method for QPCCs with finite termination based on logical Benders decomposition.
  - Proposed a tree guided master problem heuristic to search for unexplored pieces.
  - Proposed two methods for cut sparsification.

- **Future research**

---

John E. Mitchell (RPI)  Logical Benders for QCP  EUROPT, July 12, 2017
Summary & Future Research

**Summary**
- Presented a global optimization method for QPCCs with finite termination based on logical Benders decomposition.
- Proposed a tree guided master problem heuristic to search for unexplored pieces.
- Proposed two methods for cut sparsification.

**Future research**
- Extend to larger problems.
Summary & Future Research

Summary

- Presented a global optimization method for QPCCs with finite termination based on logical Benders decomposition.
- Proposed a tree guided master problem heuristic to search for unexplored pieces.
- Proposed two methods for cut sparsification.

Future research

- Extend to larger problems.
- Improve bounds during tree sparsification.
Summary & Future Research

**Summary**
- Presented a global optimization method for QPCCs with finite termination based on logical Benders decomposition.
- Proposed a tree guided master problem heuristic to search for unexplored pieces.
- Proposed two methods for cut sparsification.

**Future research**
- Extend to larger problems.
- Improve bounds during tree sparsification.
- Extend tree guided master problem for binary variables.