Logical Benders Decomposition for Quadratic Programs with Complementarity Constraints and Binary Variables[1]

John E. Mitchell[2]
Francisco Jara-Moroni[3] Jong-Shi Pang[4] Andreas Wächter[3]

[2]Rensselaer Polytechnic Institute
[3]Northwestern University
[4]University of Southern California

EUROPT, Montreal
July 12, 2017

[1]Supported by AFOSR and NSF.

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

Outline

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.

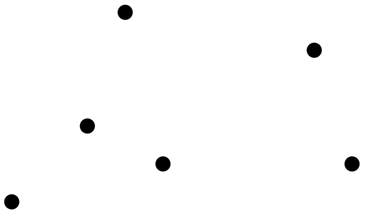
For example:
A facility location problem, with multiple competing shippers.

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)

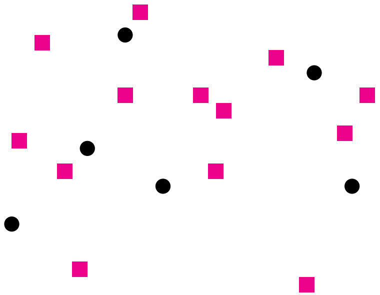
Possible locations for fuel processing.
Fixed cost for opening:
binary variables.
Limited capacity.



Biofuel supply chain design (Bai, Ouyang, Pang)

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

Farm locations.

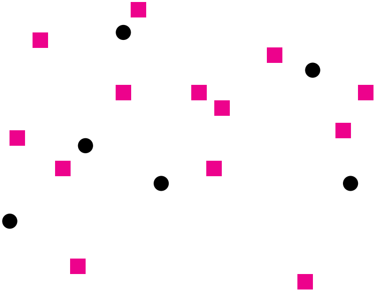


Biofuel supply chain design (Bai, Ouyang, Pang)

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.

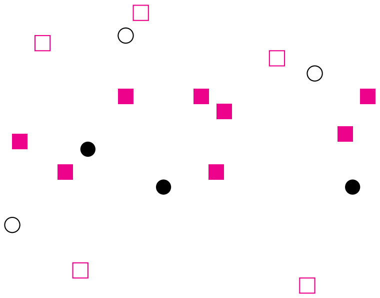


Biofuel supply chain design (Bai, Ouyang, Pang)

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:

$$\begin{aligned} \min_{x,y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0 \\ & y \text{ binary} \\ & x \in \operatorname{argmin}_x \{h(x) : \\ & \qquad r(x) \le v(y)\} \end{aligned}$$

Bilevel programs

Bilevel program:

$$\begin{aligned} \min_{x,y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0 \\ & y \text{ binary} \\ & x \in \operatorname{argmin}_x \{h(x) : r(x) \le v(y)\} \end{aligned}$$

DC-MPCC:

$$\begin{aligned} \min_{x,y,z} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0 \\ & y \text{ binary} \\ & \nabla h(x) + \nabla r(x) z = 0 \\ & 0 \le v(y) - r(x) \perp z \ge 0 \end{aligned}$$

Bilevel programs

Bilevel program:

$$\begin{aligned} \min_{x,y} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0 \\ & y \text{ binary} \\ & x \in \operatorname{argmin}_x \{ h(x) : \\ & \qquad r(x) \le v(y) \} \end{aligned}$$

DC-MPCC:

$$\begin{aligned} \min_{x,y,z} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0 \\ & y \text{ binary} \\ & \nabla h(x) + \nabla r(x) z = 0 \\ & 0 \le v(y) - r(x) \perp z \ge 0 \end{aligned}$$

Equivalent if $h(x)$ and $r(x)$ are convex, and a constraint qualification holds for the subproblems.

Our Standard Form Optimization Problem

$$\begin{array}{rrcl} \min_{x,y,w} & g^T x + x^T Q x & & \\ \text{s.t.} & A_I\, x + B_I\, y + C_I\, w & \geq & b_I \\ & A_E x + B_E y + C_E w & = & b_E \\ & 0 \leq y \perp w \geq 0 & & \end{array} \tag{QPCC}$$

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

Our Standard Form Optimization Problem

$$\begin{array}{rrcl} \min\limits_{x,y,w} & g^T x + x^T Q x & & \\ \text{s.t.} & A_I\, x + B_I\, y + C_I\, w & \geq & b_I \\ & A_E x + B_E y + C_E w & = & b_E \\ & 0 \leq y \perp w \geq 0 & & \end{array} \qquad \text{(QPCC)}$$

- $y, w \in \mathbb{R}^{n_c}$ 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

$$
\begin{array}{rrcl}
\min\limits_{x,y,w} & g^T x + x^T Q x & & \\
\text{s.t.} & A_I\, x + B_I\, y + C_I\, w & \geq & b_I \\
 & A_E x + B_E y + C_E w & = & b_E \\
 & 0 \leq y \perp w \geq 0 & &
\end{array}
\tag{QPCC}
$$

- $y, w \in \mathbb{R}^{n_c}$ and $Q \in \mathbb{R}^{n_x \times n_x}$ is a positive semi-definite matrix.
- Non-convex, disjunctive problem.
- We will discuss the addition of binary variables later.

Related work

- Local solutions

Related work

- Local solutions
  - NLP formulations: e.g. Filter-MPEC (Fletcher et al., 2004), KNITRO (Byrd et al., 2006).

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
  - Branch-and-bound (Bard & Moore, 1990; Yu, Mitchell & Pang, 2016).

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
  - Branch-and-bound (Bard & Moore, 1990; Yu, Mitchell & Pang, 2016).
- 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
  - Branch-and-bound (Bard & Moore, 1990; Yu, Mitchell & Pang, 2016).
- 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 (Sholtes, 2000; Ralph & Wright, 2004).
- Global solutions
  - Branch-and-bound (Bard & Moore, 1990; Yu, Mitchell & Pang, 2016).
- 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
  - Branch-and-bound (Bard & Moore, 1990; Yu, Mitchell & Pang, 2016).
- 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
  - Branch-and-bound (Bard & Moore, 1990; Yu, Mitchell & Pang, 2016).
- 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 $p \in \{0, 1\}^{n_c}$, the p-piece **primal** problem is

Primal Piece

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

$$
\begin{array}{rllrcll}
\phi_P(p) & \triangleq & \min\limits_{x;y;w} & g^T x + \frac{1}{2} x^T Q x & & & \\
& & \text{s.t.} & A_I\, x + B_I\, y + C_I\, w & \leq & b_I & (\ \mu_I\ ) \\
& & & A_E x + B_E y + C_E w & = & b_E & (\ \mu_E\ ) \\
& & & w_i & \leq & 0,\ i : p_i = 0 & (\ \lambda_i^w\ ) \\
& & & y_i & \leq & 0,\ i : p_i = 1 & (\ \lambda_i^y\ ) \\
& & & w, y & \geq & 0. &
\end{array}
$$

Primal Piece

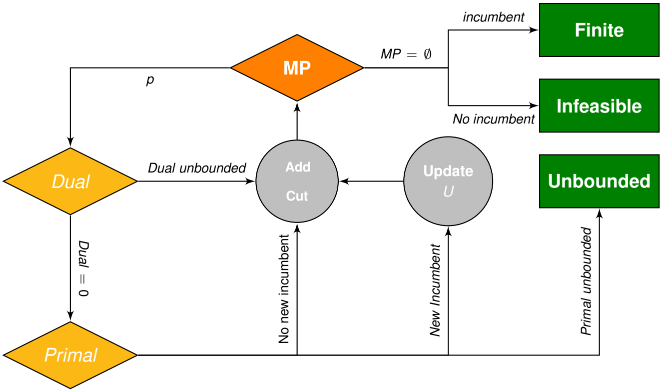
Given a binary variable $p \in \{0,1\}^{n_c}$, the p-piece **primal** problem is

$$\begin{array}{llrcll}
\phi_P(p) \triangleq & \min\limits_{x;y;w} & g^T x + \frac{1}{2} x^T Q x & & & \\
& \text{s.t.} & A_I\, x + B_I\, y + C_I\, w & \le & b_I & (\,\mu_I\,) \\
& & A_E x + B_E y + C_E w & = & b_E & (\,\mu_E\,) \\
& & w_i & \le & 0,\ i : p_i = 0 & (\,\lambda_i^w\,) \\
& & y_i & \le & 0,\ i : p_i = 1 & (\,\lambda_i^y\,) \\
& & w, y & \ge & 0. &
\end{array}$$

Therefore, (QPCC) is equivalent to solving $\min\limits_{p \in \{0,1\}^{n_c}} \phi_P(p)$.

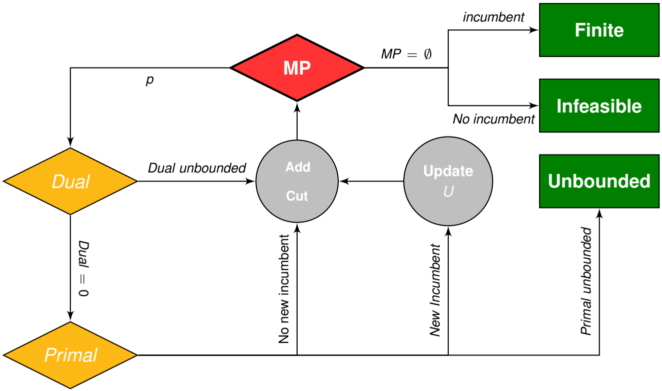
Outline

Logical Benders Decomposition on QPCC - Simplified Algorithm



Outline

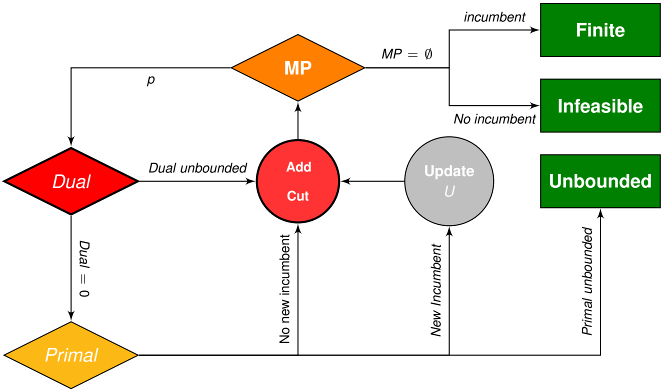
Logical Benders Decomposition on QPCC - Simplified Algorithm



Complementarity pieces p are selected in a master problem.

Outline

Logical Benders Decomposition on QPCC - Simplified Algorithm

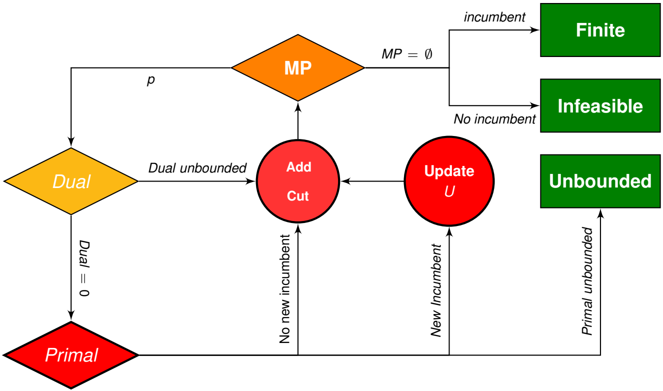


If the dual piece is unbounded, add cut to discard piece by infeasibility.
Cuts have the form

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

Outline

Logical Benders Decomposition on QPCC - Simplified Algorithm

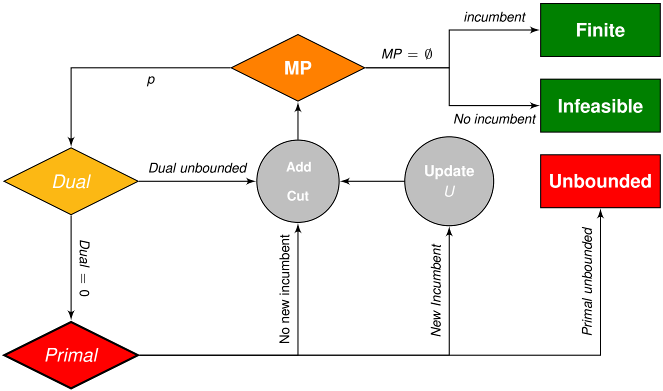


If primal is finite, update incumbent and add cut to discard piece.
Cuts also have the form

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

Outline

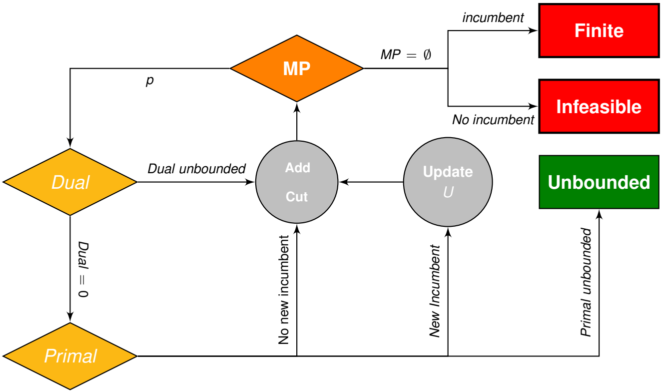
Logical Benders Decomposition on QPCC - Simplified Algorithm



If primal is unbounded, QPCC is unbounded. STOP!

Outline

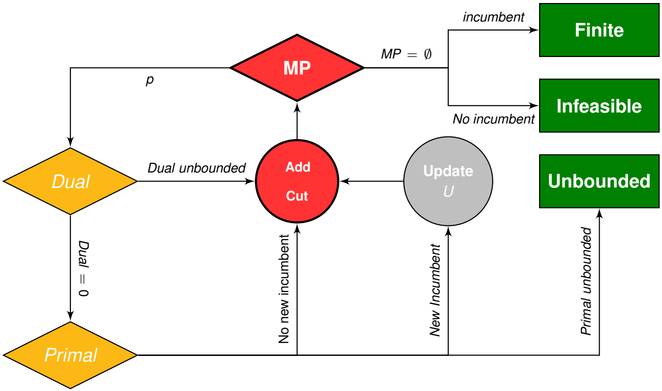
Logical Benders Decomposition on QPCC - Simplified Algorithm



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

How to select p?

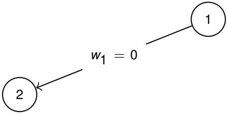
- Suppose we are solving QPCC via a **Branch-and-Bound** approach.

How to select p?

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

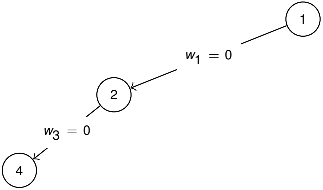
How to select p?

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



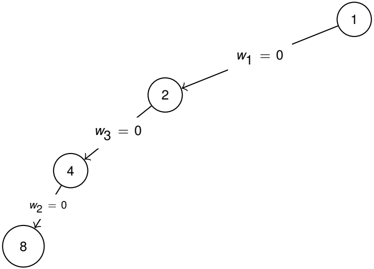
How to select p?

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



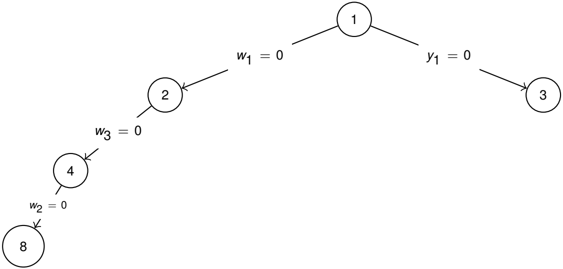
How to select *p*?

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



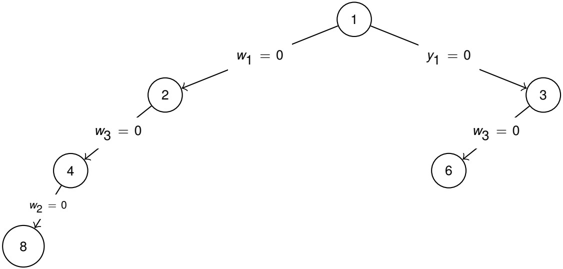
How to select p?

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



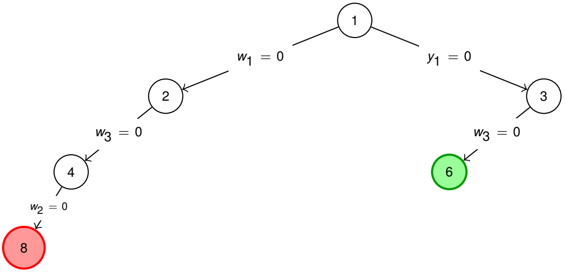
How to select p?

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



How to select p?

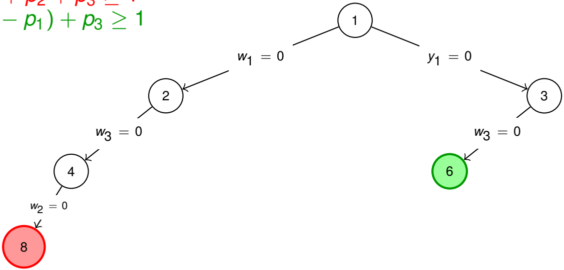
- Suppose we are solving QPCC via a **Branch-and-Bound** approach.
- Two nodes have been fathomed.



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.

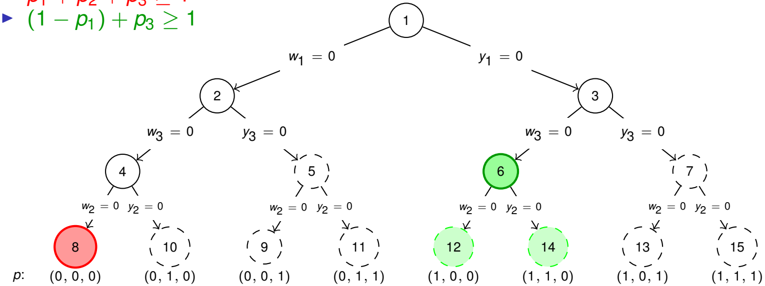
  (Recall: $p_i = 0 \leftrightarrow w_i = 0$, $p_i = 1 \leftrightarrow y_i = 0$)
  - $p_1 + p_2 + p_3 \geq 1$
  - $(1 - p_1) + p_3 \geq 1$



How to select p?

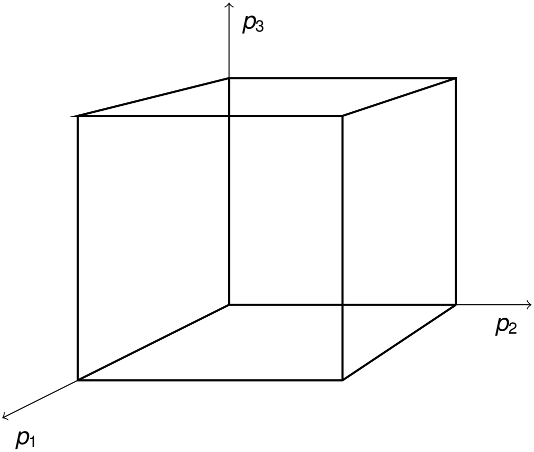
- Suppose we are solving QPCC via a **Branch-and-Bound** approach.
- Two nodes have been fathomed.
- The fathomed nodes may be interpreted as cuts.

  (Recall: $p_i = 0 \leftrightarrow w_i = 0$, $p_i = 1 \leftrightarrow y_i = 0$)
  - $p_1 + p_2 + p_3 \geq 1$
  - $(1 - p_1) + p_3 \geq 1$

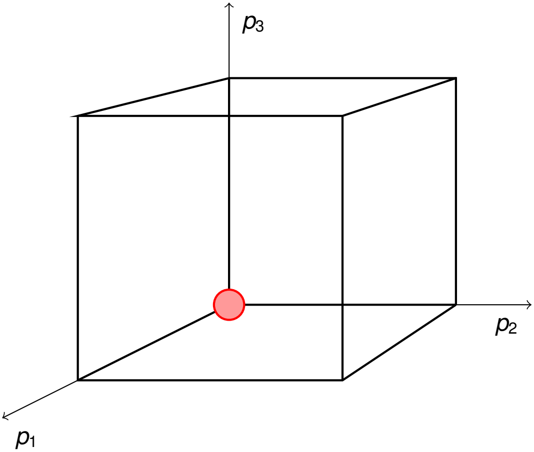


Visualizing the cuts in the Master Problem

Visualizing the cuts in the Master Problem

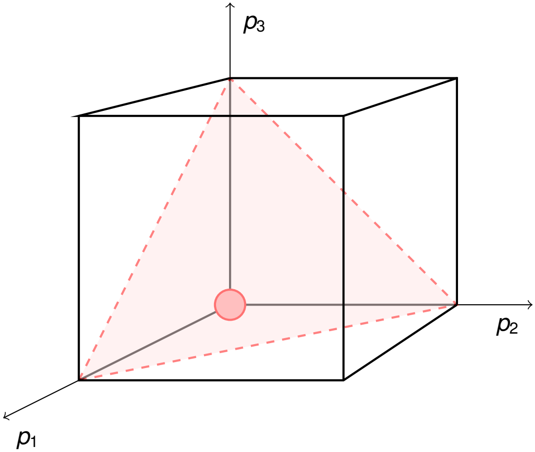


Visualizing the cuts in the Master Problem



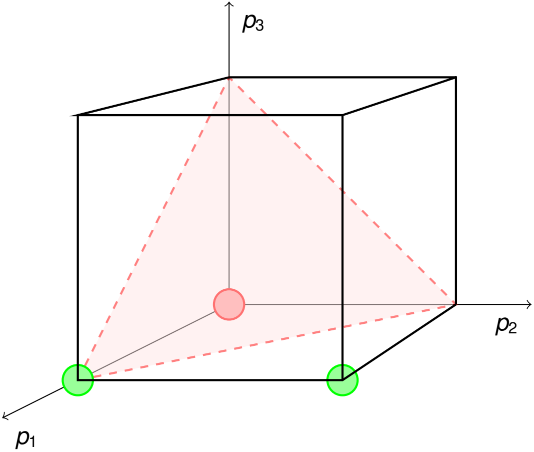
$p = (0, 0, 0)$

Visualizing the cuts in the Master Problem



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

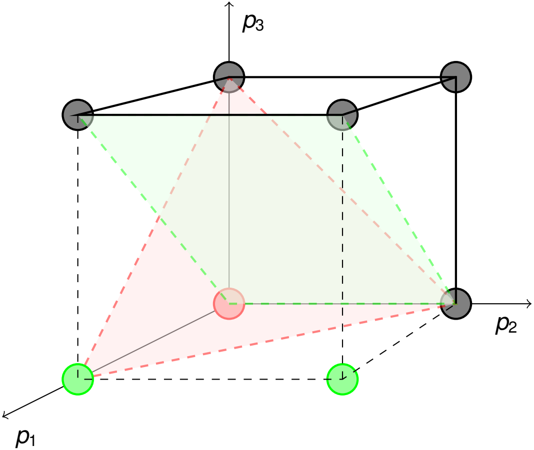
Visualizing the cuts in the Master Problem



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

$p = (1, \#, 0)$

Visualizing the cuts in the Master Problem

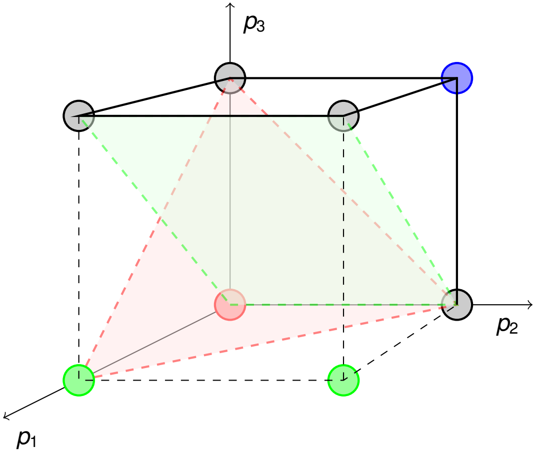


$p = (0, 0, 0)$
$p_1 + p_2 + p_3 \geq 1$

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

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 $\hat{p}$, a path P.

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

$$p_i = 0 \leftrightarrow w_i = 0$$
$$p_i = 1 \leftrightarrow y_i = 0$$

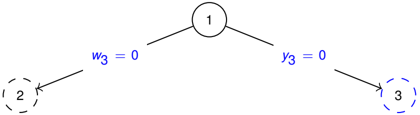
Outline of Master Problem heuristic

Input: A set of cuts $\{C_k^w, C_k^y\}$.
Output: A leaf $\hat{p}$, a path P.

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

$p_i = 0 \leftrightarrow w_i = 0$
$p_i = 1 \leftrightarrow y_i = 0$

- Branch on most explored component $\hat{\jmath}$.
  $\hat{\jmath} = \arg\max_j \left|\{k : j \in C_k^w \cup C_k^y\}\right|$



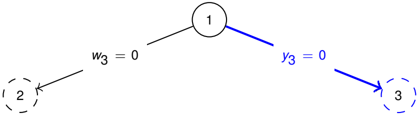
Outline of Master Problem heuristic

Input: A set of cuts $\{C_k^w, C_k^y\}$.
Output: A leaf $\hat{p}$, a path P.

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

$p_i = 0 \leftrightarrow w_i = 0$
$p_i = 1 \leftrightarrow y_i = 0$

- Branch on most explored component $\hat{\jmath}$.
- Choose least explored side of branch.
  $\hat{p}_{\hat{\jmath}} = 0$ if $|\{k : \hat{\jmath} \in C_k^w\}| \leq \left|\{k : \hat{\jmath} \in C_k^y\}\right|$; $\hat{p}_{\hat{\jmath}} = 1$ otherwise.



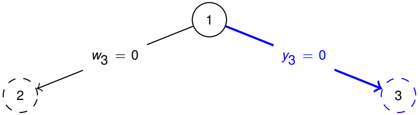
Outline of Master Problem heuristic

Input: A set of cuts $\{C_k^w, C_k^y\}$.
Output: A leaf $\hat{p}$, a path P.

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

$p_i = 0 \leftrightarrow w_i = 0$
$p_i = 1 \leftrightarrow y_i = 0$

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



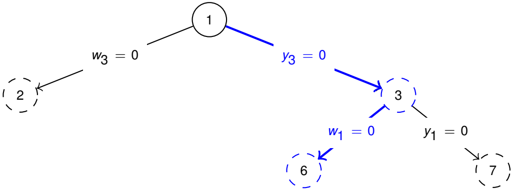
Outline of Master Problem heuristic

Input: A set of cuts $\{C_k^w, C_k^y\}$.
Output: A leaf $\hat{p}$, a path P.

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

$p_i = 0 \leftrightarrow w_i = 0$
$p_i = 1 \leftrightarrow y_i = 0$

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



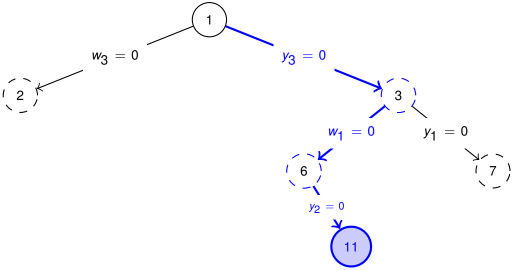
Outline of Master Problem heuristic

Input: A set of cuts $\{C_k^w, C_k^y\}$.
Output: A leaf $\hat{p}$, a path P.

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

$p_i = 0 \leftrightarrow w_i = 0$
$p_i = 1 \leftrightarrow y_i = 0$

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



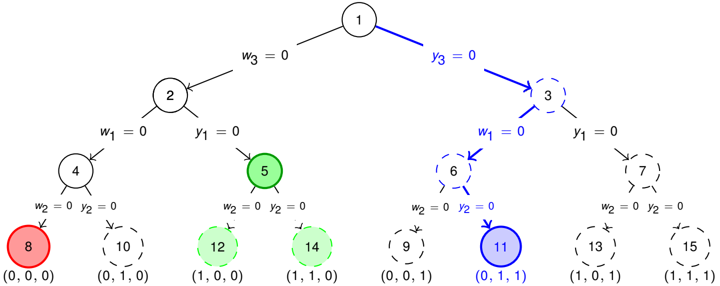
Outline of Master Problem heuristic

Input: A set of cuts $\{C_k^w, C_k^y\}$.
Output: A leaf $\hat{p}$, a path P.

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

$p_i = 0 \leftrightarrow w_i = 0$
$p_i = 1 \leftrightarrow y_i = 0$

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



Resolution

We have the two cuts:

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

$$p_i = 0 \leftrightarrow w_i = 0$$
$$p_i = 1 \leftrightarrow y_i = 0$$

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:

$$p_i = 0 \leftrightarrow w_i = 0$$
$$p_i = 1 \leftrightarrow y_i = 0$$

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

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:

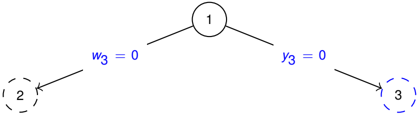
$$p_1 + p_2 + p_3 \geq 1, \qquad (1 - p_1) + p_3 \geq 1$$

$p_i = 0 \leftrightarrow w_i = 0$
$p_i = 1 \leftrightarrow y_i = 0$

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:

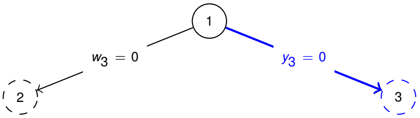
$$p_i = 0 \leftrightarrow w_i = 0$$
$$p_i = 1 \leftrightarrow y_i = 0$$

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

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:

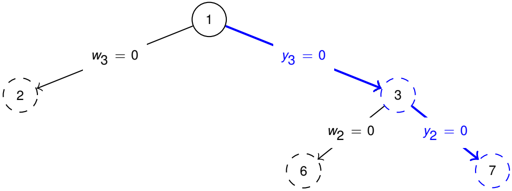
$$p_1 + p_2 + p_3 \geq 1, \qquad (1 - p_1) + p_3 \geq 1$$

$$p_i = 0 \leftrightarrow w_i = 0$$
$$p_i = 1 \leftrightarrow y_i = 0$$

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:

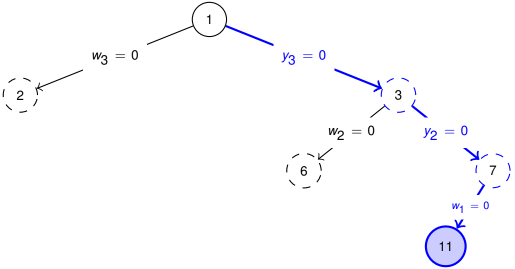
$$p_1 + p_2 + p_3 \geq 1, \qquad (1 - p_1) + p_3 \geq 1$$

$p_i = 0 \leftrightarrow w_i = 0$
$p_i = 1 \leftrightarrow y_i = 0$

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:

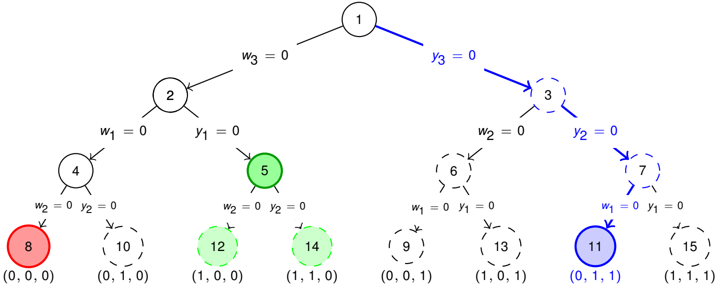
$$p_1 + p_2 + p_3 \geq 1, \qquad (1 - p_1) + p_3 \geq 1$$

$$p_i = 0 \leftrightarrow w_i = 0$$
$$p_i = 1 \leftrightarrow y_i = 0$$

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

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

$$\begin{array}{rlrcll} \phi_P(C,p) = & \min\limits_{x,y,w} & g^T x + x^T Q x & & & \\ & \text{s.t.} & A_I\, x + B_I\, y + C_I\, w & \leq & b_I, & \\ & & A_E x + B_E y + C_E w & = & b_E, & \\ & & w_i & \leq & 0 & i \in C^w \\ & & y_i & \leq & 0 & i \in C^y \\ & & w, y & \geq & 0. & \end{array}$$

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

$$\begin{array}{rrcll} \phi_P(C,p) = & \min_{x,y,w} & g^T x + x^T Q x & & \\ & \text{s.t.} \quad A_I\, x + B_I\, y + C_I\, w & \leq & b_I, & \\ & A_E x + B_E y + C_E w & = & b_E, & \\ & w_i & \leq & 0 & i \in C^w \\ & y_i & \leq & 0 & i \in C^y \\ & w, y & \geq & 0. & \end{array}$$

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

Weighted ℓ_1-norm sparsification (WL1)

- Formulate an ℓ_0-norm dual problem

$$
\begin{array}{rlrl}
\phi_{D_s}(p) = & \min\limits_{\mu_I,\, \mu_E,\, \lambda^w,\, \lambda^y} & \left\|\lambda^w\right\|_0 + \left\|\lambda^y\right\|_0 & \\
& \text{s.t.} & -A_I^T\mu_I + A_E^T\mu_E & = 0 \\
& & -B_I^T\mu_I + B_E^T\mu_E - \lambda^y & \leq 0 \\
& & -C_I^T\mu_I + C_E^T\mu_E - \lambda^w & \leq 0 \\
& & p^T\lambda^w + (\mathbf{1} - p)^T\lambda^y & = 0 \\
& & -b_I^T\mu_I + b_E^T\mu_E & = 1 \\
& & \mu_I,\, \lambda^w,\, \lambda^y & \geq 0.
\end{array}
$$

- Look for sparse λ, 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 ℓ_1-norm sparsification (WL1)

- Formulate an ℓ_1-norm dual problem

$$
\begin{aligned}
\phi_{D_s}(p) = \min_{\mu_I,\, \mu_E,\, \lambda^w,\, \lambda^y} \quad & \left\|\lambda^w\right\|_1 + \left\|\lambda^y\right\|_1 \\
\text{s.t.} \quad & -A_I^T \mu_I + A_E^T \mu_E = 0 \\
& -B_I^T \mu_I + B_E^T \mu_E - \lambda^y \le 0 \\
& -C_I^T \mu_I + C_E^T \mu_E - \lambda^w \le 0 \\
& p^T \lambda^w + (\mathbf{1} - p)^T \lambda^y = 0 \\
& -b_I^T \mu_I + b_E^T \mu_E = 1 \\
& \mu_I,\, \lambda^w,\, \lambda^y \ge 0.
\end{aligned}
$$

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

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

Weighted ℓ_1-norm sparsification (WL1)

- Formulate an ℓ_1-norm dual problem

$$
\begin{aligned}
\phi_{D_s}(p) = \min_{\mu_I,\,\mu_E,\,\lambda^w,\,\lambda^y} \quad & \sum_i (\omega_i^k \lambda_i^w + \gamma_i^k \lambda_i^y) \\
\text{s.t.} \quad & -A_I^T \mu_I + A_E^T \mu_E = 0 \\
& -B_I^T \mu_I + B_E^T \mu_E - \lambda^y \le 0 \\
& -C_I^T \mu_I + C_E^T \mu_E - \lambda^w \le 0 \\
& p^T \lambda^w + (\mathbf{1} - p)^T \lambda^y = 0 \\
& -b_I^T \mu_I + b_E^T \mu_E = 1 \\
& \mu_I,\, \lambda^w,\, \lambda^y \ge 0.
\end{aligned}
$$

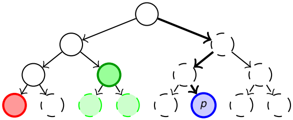
Remarks

- To solve ℓ_1 we use a weighted iterative procedure.

$$
\omega_i^{k+1} = \frac{1}{\max\{\lambda_{i,k+1}^w, \varepsilon\}} \text{ and } \gamma_i^{k+1} = \frac{1}{\max\{\lambda_{i,k+1}^y, \varepsilon\}}
$$

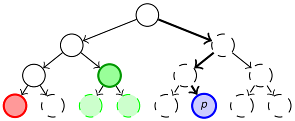
Tree guided sparsification (Tree)

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



Tree guided sparsification (Tree)

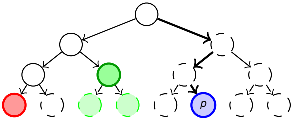
We have piece p and path P.



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.

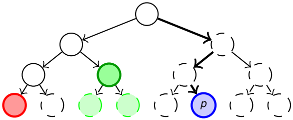


Idea: Relax complementarities of primal piece p, one by one, following the path P.

- 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.

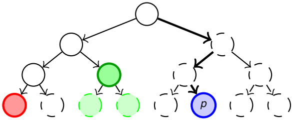


Idea: Relax complementarities of primal piece p, one by one, following the path P.

- From the example we had $p = (0, 1, 1)$ and $P = 3 \to 1 \to 2$ (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.

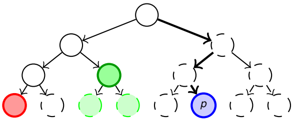


Idea: Relax complementarities of primal piece p, one by one, following the path P.

- From the example we had $p = (0, 1, 1)$ and $P = 3 \to 1 \to 2$ (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).

Tree guided sparsification (Tree)

We have piece p and path P.

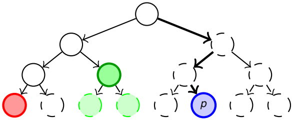


Idea: Relax complementarities of primal piece p, one by one, following the path P.

- 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.

Tree guided sparsification (Tree)

We have piece p and path P.

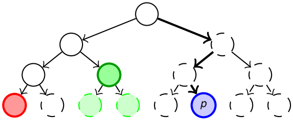


Idea: Relax complementarities of primal piece p, one by one, following the path P.

- 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).

Tree guided sparsification (Tree)

We have piece p and path P.

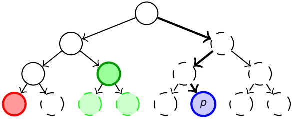


Idea: Relax complementarities of primal piece p, one by one, following the path P.

- 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 $P = 3 \to 1 \to 2$ (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.

Resulting cut is less likely to contain complementarities that are closer to the leaf p.

Hybrid Sparsification (Hyb)

- Advantages & disadvantages

Hybrid Sparsification (Hyb)

- Advantages & disadvantages
  - ℓ_1-norm sparsification

Hybrid Sparsification (Hyb)

- Advantages & disadvantages
  - ℓ_1-norm sparsification
    - Advantage: Solve few LPs ($\sim$ 6 WL1 iterations for $n_c = 100$).

Hybrid Sparsification (Hyb)

- Advantages & disadvantages
  - ℓ_1-norm sparsification
    - Advantage: Solve few LPs ($\sim$ 6 WL1 iterations for $n_c = 100$).
    - Disadvantage: Cut may not be minimal.

Hybrid Sparsification (Hyb)

- Advantages & disadvantages
  - ℓ_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
  - ℓ_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
  - ℓ_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
  - ℓ_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**

Hybrid Sparsification (Hyb)

- Advantages & disadvantages
  - ℓ_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 ℓ_1-norm sparsification. Obtain C^w and C^y.

Hybrid Sparsification (Hyb)

- Advantages & disadvantages
  - ℓ_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 ℓ_1-norm sparsification. Obtain C^w and C^y.
  - Then apply tree guided sparsification only over remaining elements of C.

Split the problem

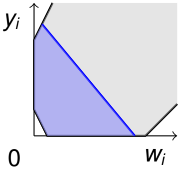
Upper bound helps getting tighter cuts.
Would like good starting upper bound U.

Split the problem

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

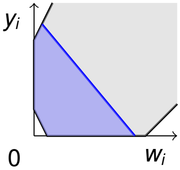
Split the problem

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



Split the problem

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



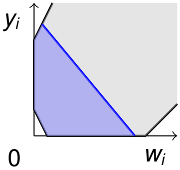
- Bounded feasible region, solve as MIQP.

  Add constraints $w \leq Tp$, $y \leq T(\mathbf{1} - p)$, 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:



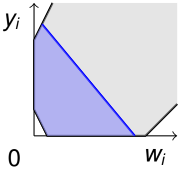
- Bounded feasible region, solve as MIQP.

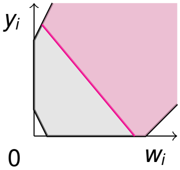
  Add constraints $w \leq Tp$, $y \leq T(\mathbf{1} - p)$, p binary, and $y_i + w_i \leq T$ for all i.

  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:





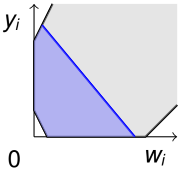
- Bounded feasible region, solve as MIQP.

  Add constraints $w \leq Tp$, $y \leq T(\mathbf{1} - p)$, 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

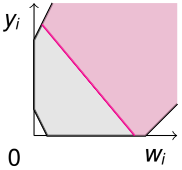
Upper bound helps getting tighter cuts.
Would like good starting upper bound U.
Pick large constant T and solve two QPCCs:



- Bounded feasible region, solve as MIQP.

  Add constraints $w \leq Tp$, $y \leq T(\mathbf{1} - p)$, p binary, and $y_i + w_i \leq T$ for all i.

  Optimal value gives **upper bound** U, with a corresponding piece.



- A region bounded away from the origin.

  Add constraint $\mathbf{1}^T y + \mathbf{1}^T w \geq T$.

  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

Numerical Results

- LPCC:

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

MIQP preprocessing, $T = 100$.

Numerical Results

- LPCC:

n_c	Total Iterations WL1	Hyb	Tree
100	34	34	33
	7	7	5
	2	2	2
	29	16	24
	20	18	31
	5	5	4
	88	75	92
	13	10	10
	120	91	97
	25	34	42
300	107	171	174
	82	63	51
	1391	1113	1047
	100	235	234
	271	216	251
	150	> 3000	> 3000
	200	170	197
	115	147	212
	378	255	271
	101	148	194

MIQP preprocessing, $T = 100$.

Numerical Results

- QPCC

n_c	Total Iterations			
	No outer box		outer Box	
	SAT	Tree	SAT	Tree
50	17	15	17	15
	38	22	2	5
	34	17	4	4
	30	19	8	3
	30	20	6	4
	37	17	25	12
	$>$100	37	$>$100	23
	$>$100	15	18	14
	38	23	38	23
	$>$100	8	8	5
100	137	35	130	34
	-	62	-	62
	102	23	102	15
	66	37	66	37
	7	6	7	5
	-	16	-	16
	-	8	-	6
	-	34	-	34
	47	3	47	3
	-	37	-	20

MIQP preprocessing, Hybrid, $T = 1000$.

Outline

Binary Variables

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

Binary Variables

- The previous settings can be extended to a binary vector z $(0 \leq z \perp \mathbf{1} - z \geq 0)$.
- 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 $(0 \leq z \perp \mathbf{1} - z \geq 0)$.
- 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 $(0 \leq z \perp \mathbf{1} - z \geq 0)$.
- 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 $(0 \leq z \perp \mathbf{1} - z \geq 0)$.
- 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 $\bar{g}, \bar{h} \leq 1$.

- Both ℓ_1 and tree-guided sparsification remain the same over this extension.

Outline

Summary & Future Research

- **Summary**

Summary & Future Research

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

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.

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**

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.