Outline

Products in Pseudo Boolean, SAT, and MIP

Bilinear Programming and McCormick relaxation
  • Special case: products that involve a binary variable

Detecting linearization of products with binary variable in MILP formulations

Exploiting product relations for cutting planes

General case: products of arbitrary variables
  • Non-convex MIQCP
"And" constraint for binary variables

\[ z = x \land y \]

Pseudo Boolean Programming
- Products of binary variables are core concept

SAT
- Products can be modeled as a set of three clauses

\[
\begin{align*}
  x \lor \bar{z} & \Leftrightarrow x = 0 \rightarrow z = 0 \\
  y \lor \bar{z} & \Leftrightarrow y = 0 \rightarrow z = 0 \\
  \bar{x} \lor \bar{y} \lor z & \Leftrightarrow x = y = 1 \rightarrow z = 1
\end{align*}
\]
Polynomials in Pseudo Boolean Programming and SAT

General "And" constraint for binary variables

\[ z = x_1 \wedge \cdots \wedge x_n \]

Pseudo Boolean Programming

- Polynomials of binary variables are core concept

SAT

- Products of \( n \) binary variables can be modeled as a set of \( n + 1 \) clauses

\[
\begin{align*}
  x_1 & \vee \bar{z} & \iff & x_1 = 0 & \rightarrow & z = 0 \\
  \ldots & & & \ldots & & \ldots \\
  x_n & \vee \bar{z} & \iff & x_n = 0 & \rightarrow & z = 0 \\
  \bar{x}_1 & \vee \ldots \vee \bar{x}_n & \vee z & \iff & x_1 = \cdots = x_n = 1 & \rightarrow & z = 1
\end{align*}
\]
A Mixed Integer Quadratically Constraint Program (MIQCP) is defined as

\[
\begin{align*}
\text{min} & \quad c^T x + x^T Q_0 x \\
\text{s.t.} & \quad a_i^T x + x^T Q_i x \leq b_i \\
& \quad a_m^T x + x^T Q_m x \leq b_m \\
& \quad l \leq x \leq u \\
& \quad x_j \in \mathbb{Z} \quad \text{for all } j \in I
\end{align*}
\]

- \( Q_k \) are symmetric matrices
- If all \( Q_k \) are positive semi-definite, then QCP relaxation is convex
  - MIQCPs with positive semi-definite \( Q_k \) can be solved by Gurobi since version 5.0
- For \( Q = Q_k \), any non-zero element \( Q_{ij} \neq 0 \) gives rise to a product term \( Q_{ij} x_i x_j \) in the constraint or objective
- What if quadratic constraints are non-convex?
Mixed Integer Bilinear Program

Introduce auxiliary variables

\[ z_{ij} := x_i x_j \]

for each product term \( x_i x_j \) that appears in some \( Q = Q_k \) with \( Q_{ij} \neq 0 \)

A Mixed Integer Bilinear Program is defined as

\[
\begin{align*}
\text{min} \quad & c^T x + d^T z \\
\text{s.t.} \quad & Ax + Dz \leq b \\
& -x_i x_j + z_{ij} = 0 \quad \text{for all } (i, j) \in S \\
& l \leq x \leq u \\
& x_j \in \mathbb{Z} \quad \text{for all } j \in I
\end{align*}
\]
McCormick Relaxation of Bilinear Constraints

Mixed product case: $-x_i x_j + z = 0$

McCormick lower and upper envelopes:

$$-z + l_j x_i + l_i x_j \leq l_j l_i$$
$$-z + u_j x_i + u_i x_j \leq u_j u_i$$

$$-z + u_j x_i + l_i x_j \geq u_j l_i$$
$$-z + l_j x_i + u_i x_j \geq l_j u_i$$

pictures from Costa and Liberti: "Relaxations of multilinear convex envelopes: dual is better than primal"
McCormick Relaxation: Binary Variables

McCormick relaxation

\[-z + l_j x_i + l_i x_j \leq l_j l_i \]
\[-z + u_j x_i + u_i x_j \leq u_j u_i \]
\[z - u_j x_i - l_i x_j \leq -u_j l_i \]
\[z - l_j x_i - u_i x_j \leq -l_j u_i \]

Special case for products of two binary variables $x_i, x_j \in \{0,1\}$

\[-z \leq 0 \]
\[-z + x_i + x_j \leq 1 \]
\[z - x_i \leq 0 \]
\[z - x_j \leq 0 \]
McCormick Relaxation: Binary Variables

McCormick relaxation

\[-z + l_j x_i + l_i x_j \leq l_j l_i\]
\[-z + u_j x_i + u_i x_j \leq u_j u_i\]
\[z - u_j x_i - l_i x_j \leq -u_j l_i\]
\[z - l_j x_i - u_i x_j \leq -l_j u_i\]

Special case for products of two binary variables \(x_i, x_j \in \{0,1\}\), and thus \(z \in \{0,1\}\)

\[-z \leq 0\]
\[-z + x_i + x_j \leq 1 \iff x_i = x_j = 1 \rightarrow z = 1\]
\[z - x_i \leq 0 \iff x_i = 0 \rightarrow z = 0\]
\[z - x_j \leq 0 \iff x_j = 0 \rightarrow z = 0\]
McCormick Relaxation: Binary Variables

McCormick relaxation

\[-z + l_j x_i + l_i x_j \leq l_j l_i\]
\[-z + u_j x_i + u_i x_j \leq u_j u_i\]
\[z - u_j x_i - l_i x_j \leq -u_j l_i\]
\[z - l_j x_i - u_i x_j \leq -l_j u_i\]

Special case for product of binary variable $x_i \in \{0,1\}$ and non-binary variable $x_j \in [l_j, u_j]$

\[-z + l_j x_i \leq 0 \iff x_i = 0 \rightarrow z \geq 0 \quad x_i = 1 \rightarrow z \geq l_j\]
\[-z + u_j x_i + x_j \leq u_j \iff x_i = 0 \rightarrow z \geq x_j - u_j \quad x_i = 1 \rightarrow z \geq x_j\]
\[z - u_j x_i \leq 0 \iff x_i = 0 \rightarrow z \leq 0 \quad x_i = 1 \rightarrow z \leq u_j\]
\[z - l_j x_i - x_j \leq -l_j \iff x_i = 0 \rightarrow z \leq x_j - l_j \quad x_i = 1 \rightarrow z \leq x_j\]
McCormick Relaxation: Binary Variables

McCormick relaxation

\[-z + l_j x_i + l_i x_j \leq l_j l_i\]
\[-z + u_j x_i + u_i x_j \leq u_j u_i\]
\[z - u_j x_i - l_i x_j \leq -u_j l_i\]
\[z - l_j x_i - u_i x_j \leq -l_j u_i\]

Special case for product of binary variable \(x_i \in \{0, 1\}\) and non-binary variable \(x_j \in [l_j, u_j]\)

\[-z + l_j x_i \leq 0 \iff x_i = 0 \rightarrow z \geq 0 \quad x_i = 1 \rightarrow z \geq l_j\]
\[-z + u_j x_i + x_j \leq u_j \iff x_i = 0 \rightarrow z \geq x_j - u_j \quad x_i = 1 \rightarrow z \geq x_j\]
\[z - u_j x_i \leq 0 \iff x_i = 0 \rightarrow z \leq 0 \quad x_i = 1 \rightarrow z \leq u_j\]
\[z - l_j x_i - x_j \leq -l_j \iff x_i = 0 \rightarrow z \leq x_j - l_j \quad x_i = 1 \rightarrow z \leq x_j\]
Products of Binary and Unbounded Variable

Product of \( x_i \in \{0, 1\} \) and \( x_j \in [l_j, u_j] \) with \( l_j \in \mathbb{R} \cup \{-\infty\} \) and \( u_j \in \mathbb{R} \cup \{\infty\} \)

\[
\begin{align*}
-z + l_j x_i & \leq 0 \\
z - u_j x_i & \leq 0
\end{align*}
\]

\( \iff \) \( x_i = 0 \rightarrow z = 0 \)

\( \iff \) \( SOS1(\bar{x}_i, z) \)

\[
\begin{align*}
-z + u_j x_i + x_j & \leq u_j \\
z - l_j x_i - x_j & \leq -l_j
\end{align*}
\]

\( \iff \) \( x_i = 1 \rightarrow z = x_j \)

\( \iff \) \( SOS1(x_i, s) \)

\( -\infty \leq s \leq \infty \)

Special Ordered Set Type 1

- At most one of the variables can be non-zero: \( SOS1(x_1, \ldots, x_n) \iff |\{i|x_i \neq 0\}| \leq 1 \)

Special Ordered Set Type 2

- At most two of the variables can be non-zero, and if two are non-zero they need to be adjacent
- Used to model piece-wise linear functions
- Not subject of this talk
MIQCP Linearization

At the end of presolve, Gurobi decides whether product terms with at least one binary are linearized

- Default settings: linearize most of the time
  - Except if linearization adds too many variables and constraints

If no product of two non-binary variables existed, final model is MILP

User may have already performed linearization in original model

MIP solvers often work on MILPs that model products of a binary with some other variable

- Can we rediscover the product relationship from the linear constraints?
- Can we exploit techniques from MIQCP or Bilinear Programming to solve these MILPs faster?
Products in Mixed Integer Linear Programs

Products do not need to be explicitly stated as bilinear constraints

Products of a binary variable \( x_i \) and an arbitrary variable \( x_j \) can be modeled as linear constraints

\[
\begin{align*}
  z_{ij} & \leq x_i x_j : & x_i = 1 & \Rightarrow z_{ij} \leq x_j & & z_{ij} - x_j + (U_{ij} - l_j)x_i & \leq U_{ij} - l_j \\
  & x_i = 0 & \Rightarrow z_{ij} \leq 0 & & z_{ij} - U_{ij}x_i & \leq 0 \\
  z_{ij} & \geq x_i x_j : & x_i = 1 & \Rightarrow z_{ij} \geq x_j & & -z_{ij} + x_j + (u_j - L_{ij})x_i & \leq u_j - L_{ij} \\
  & x_i = 0 & \Rightarrow z \geq 0 & & -z_{ij} + L_{ij}x_i & \leq 0
\end{align*}
\]

We can detect these structures in a MILP formulation
Useful MIP Data Structures

Implied bounds
• MIP solvers store implied bound relations between binary and non-binary variables in a table

\[
\begin{align*}
    x_i = 0 & \rightarrow x_j \leq b, \\
    x_i = 0 & \rightarrow x_j \geq b, \\
    x_i = 1 & \rightarrow x_j \leq b, \\
    x_i = 1 & \rightarrow x_j \geq b
\end{align*}
\]

Cliques
• MIP solvers store implications between binary variables in a clique table

\[
\begin{align*}
    x_i = 0 & \rightarrow x_j = 0, \\
    x_i = 0 & \rightarrow x_j = 1, \\
    x_i = 1 & \rightarrow x_j = 0, \\
    x_i = 1 & \rightarrow x_j = 1
\end{align*}
\]

\[
\begin{align*}
    -x_i + x_j & \leq 0, \\
    -x_i - x_j & \leq -1, \\
    x_i + x_j & \leq 1, \\
    x_i - x_j & \leq 0
\end{align*}
\]

• Can aggregate multiple implications into larger cliques, e.g.:

\[
    x_1 + x_2 + x_3 + x_4 + x_5 \leq 1
\]

to represent all implications \( x_i = 1 \rightarrow x_j = 0 \) for all \( i, j = 1, \ldots, 5, \ i \neq j \)
Detection of Products

More general version:

\[ x_i = 1 \rightarrow z_{ij} \leq ax_j + b \]
\[ x_i = 0 \rightarrow z \leq cx_j + d \]

We can detect these structures in a MILP formulation

1. Inspect inequalities with three variables, from which at least one is binary
2. For each binary variable in this constraint, assume this is \( x_i \) and search for the other constraint
   
   - \( c \neq 0 \): a constraint with \( x_i, x_j, \) and \( z_{ij} \)
   
   - \( c = 0, M \neq 0 \): a constraint with \( x_i \) and \( z_{ij} \) (i.e., an implied bound if \( z_{ij} \) is non-binary, a clique if \( z_{ij} \) is binary)
   
   - \( c = 0, M = 0 \): a global upper bound on \( z_{ij} \)

Result: \( z_{ij} \leq x_i(ax_j + b) + (1 - x_i)(cx_j + d) = (a - c)x_i x_j + (b - d)x_i + cx_j + d \)

If we also find a product relation in the \( \geq \) direction with the same \( a, b, c, d \): an equality relation

We store product relations as \( x_i x_j \preceq \hat{a}z_{ij} + \hat{b}x_i + \hat{c}x_j + \hat{d} \)
Detection of Products – Details

Basic algorithm:
1. Find "implied relations" $x_i = f \rightarrow z_{ij} \leq ax_j + b$ by scanning length 3 constraints with at least one binary
2. For each implied relation, try to find other constraint $x_i = 1 - f \rightarrow z_{ij} \leq cx_j + d$ as
   • implied relation (binary search for $(x_i, x_j, z_{ij})$ in implied relations table)
   • implied bound ($c = 0$, $z_{ij}$ non-binary; linear scan in implications of $x_i = 1 - f$),
   • clique ($c = 0$, $z_{ij}$ binary; scan clique table columns of $z_{ij}$ and $\bar{z}_{ij}$ for match with column of $x_i$ or $\bar{x}_i$)
   • unconditional relation (clique, implied bound, or length 2 inequalities for $x_j$ and $z_{ij}$), or
   • global bound of $z_{ij}$
3. Also try to find constraints in opposite direction to identify equality product relations

Special case: product relations defined without length 3 constraint

$$z_{ij} \leq ax_j + b \quad \text{(unconditional relation)}$$
$$x_i = 0 \rightarrow z_{ij} \leq d \quad \text{(implied bound)}$$

yields:

$$z_{ij} \leq x_i(ax_j + b) + (1 - x_i)d = ax_ix_j + (b - d)x_i + d$$

$$\iff \quad x_ix_j \geq \frac{1}{a} (d - b)x_i + \frac{1}{a}z_{ij} - \frac{d}{a}$$

Extended algorithm: also consider implied bounds in Step 1

• What about cliques? Too expensive!
• And already implicitly stored in clique table

$$z_{ij} + x_i \leq 1, z_{ij} + x_j \leq 1 \Rightarrow z_{ij} \leq (1 - x_i)(1 - x_j)$$
RLT Cuts

A Mixed Integer Bilinear Program is defined as

\[
\begin{align*}
\min & \quad c^T x + d^T z \\
\text{s.t.} & \quad Ax + Dz \leq b \\
& \quad -x_i x_j + z_{ij} = 0 \quad \text{for all } (i,j) \in S \\
& \quad x \geq 0 \\
& \quad x_j \in \mathbb{Z} \quad \text{for all } j \in I
\end{align*}
\]

RLT Cuts:
1. Multiply a linear constraint \( ax + dz \leq b \) by some variable \( x_j \geq 0 \)
2. Linearize resulting quadratic constraint by
   a) replacing products \( x_i x_j \) for \( (i,j) \in S \) by corresponding product variables \( z_{ij} \)
   b) relaxing products \( x_i x_j \) for \( (i,j) \notin S \) by linear term, e.g., McCormick underestimator

How can such a cut be violated?
• Replacing product terms \( a_i x_i x_j = a_i z_{ij} \) increases violation if \( a_i x_i^* x_j^* < a_i z_{ij}^* \)
Separation of RLT Cuts

For each variable $x_j$ consider $(x_j - l_j)$ and $(u_j - x_j)$

- For each violated product relation of $x_i x_j$ mark constraints of $x_i$ in which $a_i$ has the right sign
- Multiply each marked constraint by $(x_j - l_j)$ or $(u_j - x_j)$, respectively to yield quadratic constraint
- Substitute product terms using known product relations or a linear relaxation
- If violated: add to cut table
Substituting Product Terms

Need to get rid of product terms \( x_i x_j \) to turn quadratic into linear constraint

- Substitute \( x_j^2 = x_j \) if \( x_j \) is binary
- Tangent for \( a_j x_j^2 \) if \( x_j \) is non-binary and \( a_j > 0 \)
- Secant for \( a_j x_j^2 \) if \( x_j \) is non-binary and \( a_j < 0 \)
- Clique relationship if \( x_i \) and \( x_j \) are binary and clique exists
  - \( x_i + x_j \leq 1 \) \( \Rightarrow \) \( x_i x_j = 0 \)
  - \( x_i + (1 - x_j) \leq 1 \) \( \Rightarrow \) \( x_i x_j = x_i \)
  - \( (1 - x_i) + x_j \leq 1 \) \( \Rightarrow \) \( x_i x_j = x_j \)
  - \( (1 - x_i) + (1 - x_j) \leq 1 \) \( \Rightarrow \) \( x_i x_j = x_i + x_j - 1 \)
- Product relationship, if product relationship exists for \( x_i x_j \) in the right direction
  - Product table for binary times non-binary variable, detected in linear constraints
  - Bilinear constraints
- Else: one of the two lower (\( a_j > 0 \)) or one of the two upper (\( a_j < 0 \)) McCormick envelope constraints
  - Pick the one that is tighter for the current LP solution
Tricks to Speed-Up Separation

Check violation on constraint projected to variables that are not at bound
- McCormick constraints are tight if at least one of the variables is at a bound
- Replacing product term with product variable will not change violation of cut
- Hence:
  - Only check violation on projected system
  - Repeat substitution procedure on full constraint if violated cut has been found on projected system
- In basic LP solution, most variables are typically at one of their bounds
  - Can work on a much smaller system!

Make sure to not flood the solver’s cut table with too many RLT cuts
- It happens frequently that we can find tons of RLT cuts
- Will slow down overall MIP solving process
- Solution:
  - Add RLT cuts to local cut table
  - Filter local cut table using regular filtering based on violation and orthogonality
  - Add surviving cuts to global cut table
BQP Cuts

Boolean Quadric Polytope

- \( BQP_n = \{(x, z) \in \mathbb{R}^n \times \mathbb{R}^{n \times n} \mid z_{ij} = x_i x_j \text{ for all } i, j \in [n], x_j \in \{0, 1\} \text{ for all } j \in [n]\} \)
- BQP is equivalent to the Cut Polytope

Quadratic Programming with Box Constraints

- \( QPB_n = \{(x, z) \in \mathbb{R}^n \times \mathbb{R}^{n \times n} \mid z_{ij} = x_i x_j \text{ for all } i, j \in [n], 0 \leq x_j \leq 1 \text{ for all } j \in [n]\} \)
  - Just the continuous relaxation of \( BQP_n \)
  - Theorem (Burer and Letchford, 2008):
    
    Every valid inequality for \( BQP_n \) is also valid for \( QPB_n \)

- Hence, facets of \( BQP_n \) provide cuts for bilinear constraints with bounded variables
- Padberg (1989) studied facets of \( BQP_n \)
  - RLT inequalities are facets if \( i \neq j \)
  - PSD inequalities are facets
  - "Clique" and "Cut" inequalities are facets
  - "Cut type" inequalities are facets, generalize "Clique" and "Cut" inequalities (Yajima and Fujie, 1998)
BQP Clique Cuts

Theorem (Padberg, 1989): For any $S \subseteq \{1, \ldots, n\}$ with $|S| \geq 3$, and any $\alpha \in \{1, \ldots, |S| - 2\}$, the clique inequality

$$\alpha \sum_{i \in S} x_i - \sum_{(i,j) \subset S} z_{ij} \leq \frac{\alpha(\alpha+1)}{2}$$

is valid and facet defining.

Simple case: triangle inequalities ($\alpha = 1$, $|S| = 3$): $\sum_{i \in S} x_i - \sum_{(i,j) \subset S} z_{ij} \leq 1$

$$\begin{align*}
    x_1 &+ x_2 &- z_{12} &\leq 1 \\
    x_1 & &+ x_3 &- z_{13} &\leq 1 \\
    &x_2 &+ x_3 &- z_{23} &\leq 1
\end{align*}$$

- model inequalities for $x_i = x_j = 1 \rightarrow z_{ij} = 1$

$$\begin{align*}
    x_1 &+ x_2 &+ x_3 &- z_{12} &- z_{13} &- z_{23} &\leq 1
\end{align*}$$

zero-half cut

Facet-defining even if $S$ is not a maximal clique!
BQP Triangle Cuts

Switching (i.e., complementation of $x$ and associated changes in $z$) yields four triangle inequalities:

\[
\begin{align*}
 x_1 &+ x_2 &+ x_3 &- z_{12} &- z_{13} &- z_{23} &\leq 1 \\
-x_1 & & &+ z_{12} &+ z_{13} &- z_{23} &\leq 0 \\
-x_2 & & &+ z_{12} &- z_{13} &+ z_{23} &\leq 0 \\
-x_3 &- z_{12} &+ z_{13} &+ z_{23} &\leq 0
\end{align*}
\]

Separation

- Enumerate all triangles to collect all violated triangle cuts
- Filter cuts and pass surviving cuts to cut pool
PSD Cuts

With \( Z = \{z_{ij}\}_{i,j \in [n]} \) we have

\[
Z = xx^T \iff Z - xx^T = 0 \implies Z - xx^T \succeq 0 \iff \begin{bmatrix} 1 & x^T \\ x & Z \end{bmatrix} \succeq 0
\]

\[
\implies (s \quad v^T) \begin{bmatrix} 1 & x^T \\ x & Z \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix} \geq 0 \text{ for all } v \in \mathbb{R}^n, s \in \mathbb{R}
\]

\[
\implies v^T Z v + (2s)v^T x + s^2 \geq 0 \text{ for all } v \in \mathbb{R}^n, s \in \mathbb{R}
\]

How to find \( s \) and \( v \) to get violated cut?

- Eigenvector of \( \begin{bmatrix} 1 & x^T \\ x & Z \end{bmatrix} \) that corresponds to negative eigenvalue
Computational Results – Nodes (RLT and BQP Cuts)

MILP: 3924 models
MIQP: 309 models
MIQCP: 438 models
Non-convex MIQCP: 325 models
Total time: 338 CPU daysb
Computational Results – Time (RLT and BQP Cuts)

- 1.6% speed-up
- 8.1% speed-up
- 1.3% speed-up
- 71.5% speed-up

MILP: 3924 models
MIQP: 309 models
MIQCP: 438 models
Non-conv: 325 models
Total time: 338 CPU days