Mixed Integer Bilinear Programming

GUROBI OPTIMIZATION

The World's Fastest Solver

Tobias Achterberg

26 June 2019

Mixed Integer Quadratically Constrained Programming

A Mixed Integer Quadratically Constrained Program (MIQCP) is defined as

$$
\begin{array}{lrcll}
\min & c^T x & + & x^T Q_0 x & \\
\text{s.t.} & a_1^T x & + & x^T Q_1 x & \leq \quad b_1 \\
 & & \ldots & & \\
 & a_m^T x & + & x^T Q_m x & \leq \quad b_m \\
 & l & \leq & x & \leq \quad u \\
 & & & x_j & \in \quad \mathbb{Z} \quad \text{for all } j \in I
\end{array}
$$

- Q_k are symmetric matrices
- 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
- 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

- What if quadratic constraints are non-convex?

Mixed Integer Bilinear Program

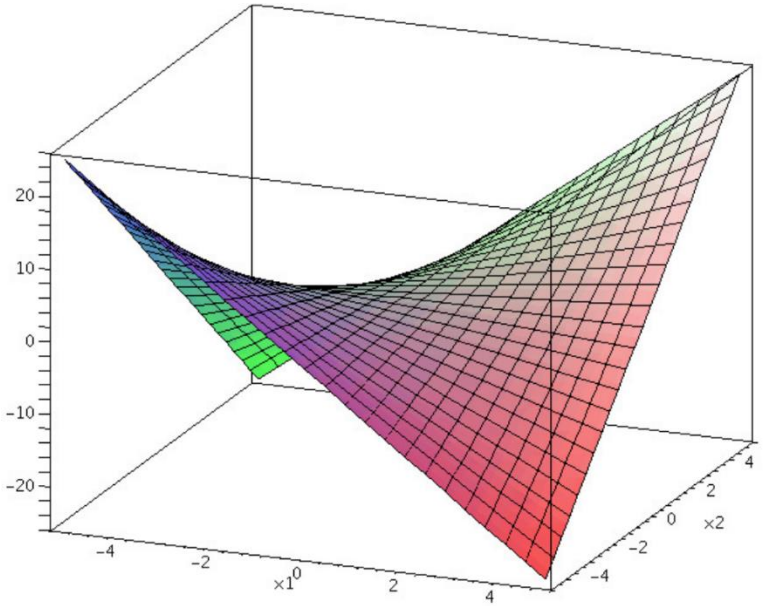
Introduce auxiliary variables

$$z_{ij} := x_i x_j$$

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

A Mixed Integer Bilinear Program is defined as

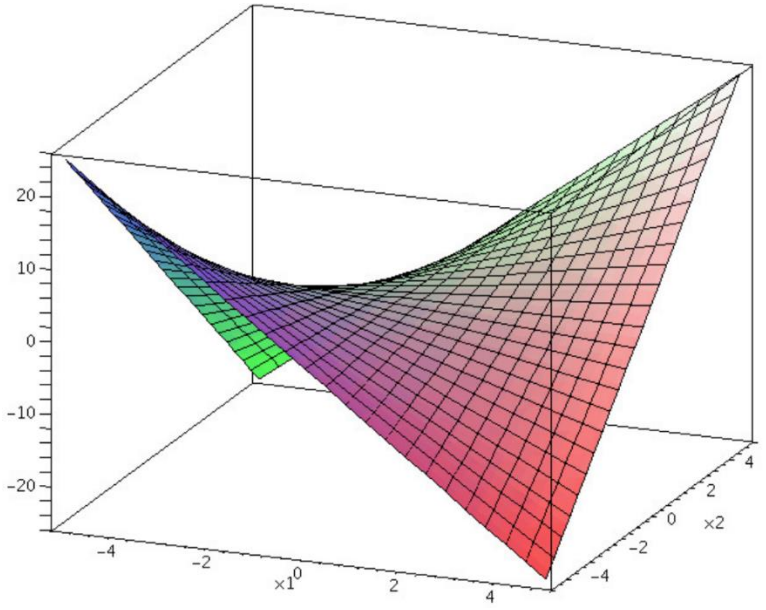
$$\begin{array}{lccccccl} \min & c^T x & + & d^T \mathrm{z} & & & & \\ \text{s.t.} & Ax & + & D\mathrm{z} & \leq & b & & \\ & -x_i x_j & + & \mathrm{z}_{ij} & = & 0 & \text{for all } (i,j) \in S & \\ & l & \leq & x & \leq & \mathrm{u} & & \\ & & & x_j & \in & \mathbb{Z} & \text{for all } j \in I & \end{array}$$



picture from Costa and Liberti: "Relaxations of multilinear convex envelopes: dual is better than primal"

McCormick Relaxation of Bilinear Constraints

Mixed product case: $-z_{ij} + x_i x_j = 0$

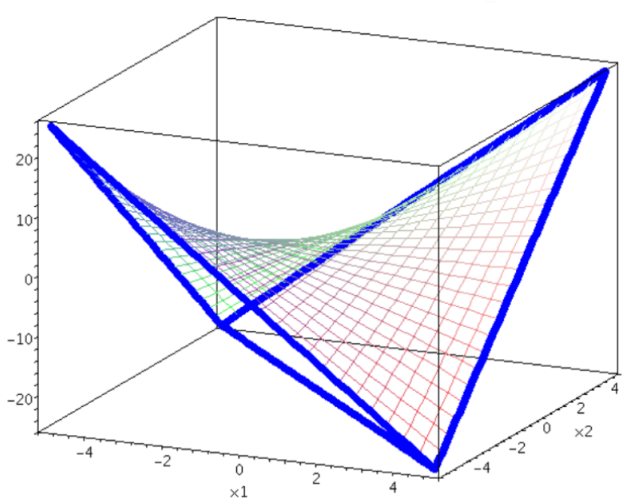


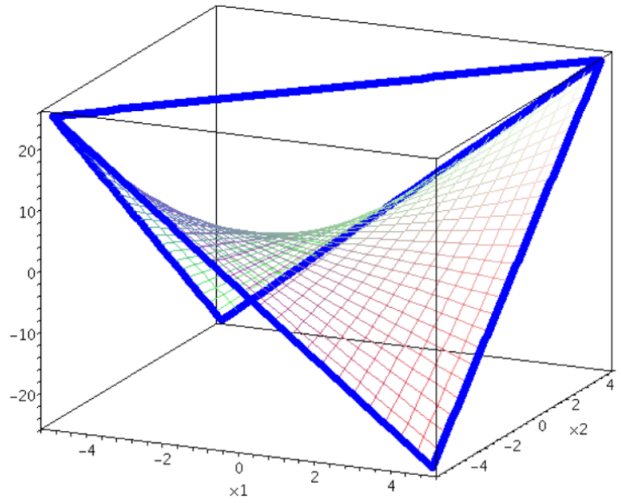
pictures from Costa and Liberti: "Relaxations of multilinear convex envelopes: dual is better than primal"

McCormick lower and upper envelopes:

$$-z_{ij} + l_j x_i + l_i x_j \leq l_j l_i$$
$$-z_{ij} + u_j x_i + u_i x_j \leq u_j u_i$$

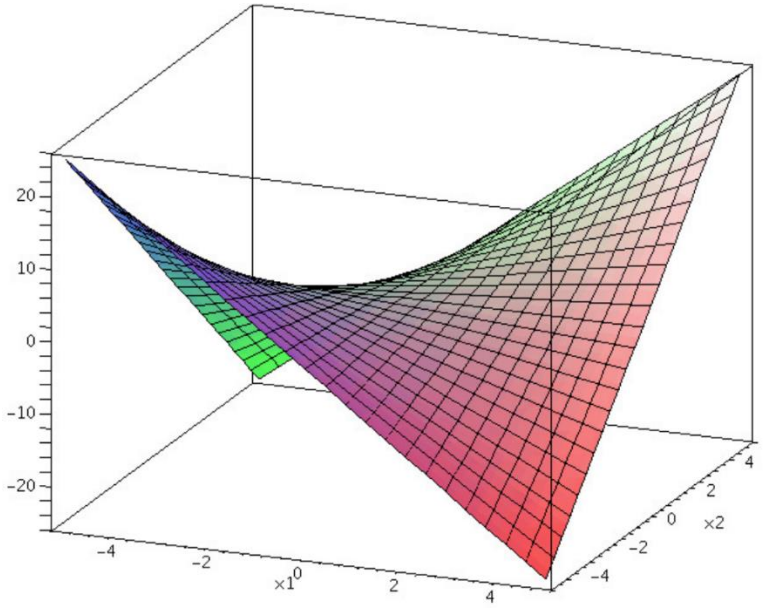
$$-z_{ij} + u_j x_i + l_i x_j \geq u_j l_i$$
$$-z_{ij} + l_j x_i + u_i x_j \geq l_j u_i$$





McCormick Relaxation of Bilinear Constraints

Mixed product case: $-z_{ij} + x_i x_j = 0$



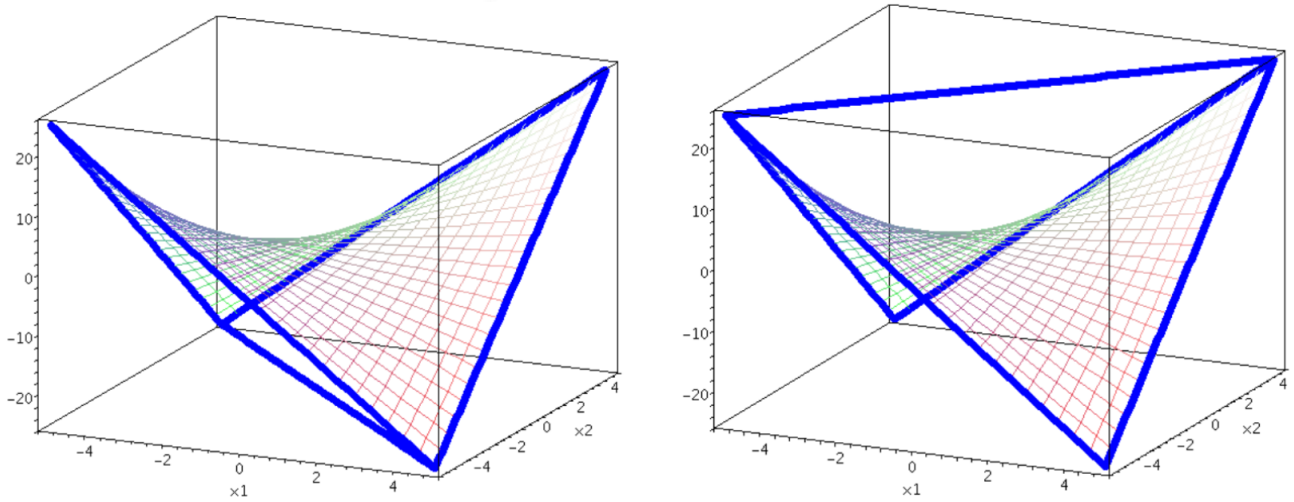
pictures from Costa and Liberti: "Relaxations of multilinear convex envelopes: dual is better than primal"

McCormick lower and upper envelopes:

$$-z_{ij} + l_j x_i + l_i x_j \leq l_j l_i$$
$$-z_{ij} + u_j x_i + u_i x_j \leq u_j u_i$$

$$-z_{ij} + u_j x_i + l_i x_j \geq u_j l_i$$
$$-z_{ij} + l_j x_i + u_i x_j \geq l_j u_i$$

coefficients depend on local bounds



Adaptive Constraints

Coefficients and right hand sides of McCormick constraints depend on local bounds of variables

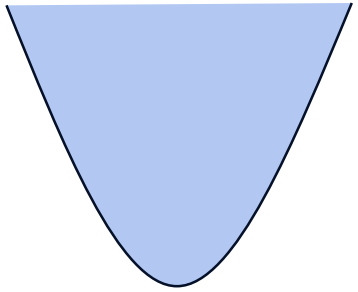
- Whenever local bounds change, LP coefficients and right hand sides are updated
- May lead to singular or ill-conditioned basis
  - In worst case, simplex needs to start from scratch

Adaptive Constraints and Branching

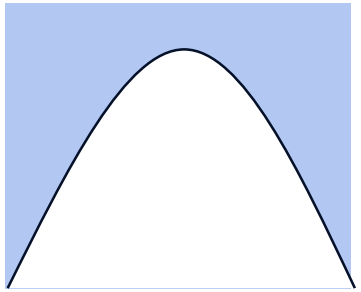
Algorithmic treatment of bilinear constraints

- General form: $a^T z + dx_i x_j \leqq b$ (linear sum plus single product term, inequality or equation)

Consider square case ($i = j$):



convex
$-z + x^2 \leq 0$



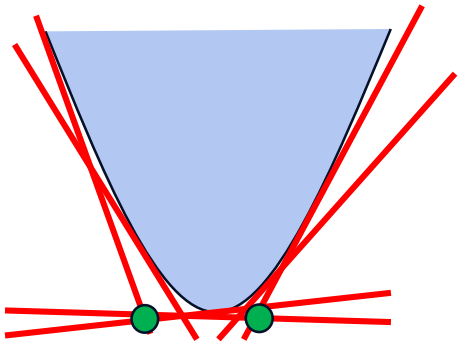
non-convex
$-z - x^2 \leq 0$

Adaptive Constraints and Branching

Algorithmic treatment of bilinear constraints

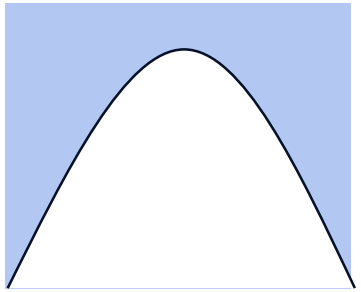
- General form: $a^T z + dx_i x_j \leqq b$ (linear sum plus single product term, inequality or equation)

Consider square case ($i = j$):



convex

$-z + x^2 \leq 0$



non-convex

$-z - x^2 \leq 0$

easy: add tangent cuts

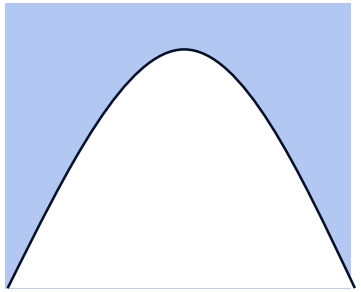
Adaptive Constraints and Branching

Algorithmic treatment of bilinear constraints

- General form: $a^T z + dx_i x_j \leqq b$ (linear sum plus single product term, inequality or equation)

Consider square case ($i = j$):

non-convex
$-z - x^2 \leq 0$



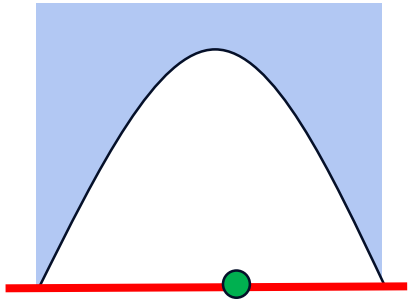
Adaptive Constraints and Branching

Algorithmic treatment of bilinear constraints

- General form: $a^T z + dx_i x_j \leqq b$ (linear sum plus single product term, inequality or equation)

Consider square case ($i = j$):

non-convex
$-z - x^2 \leq 0$



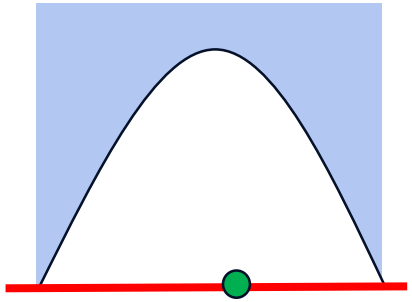
Adaptive Constraints and Branching

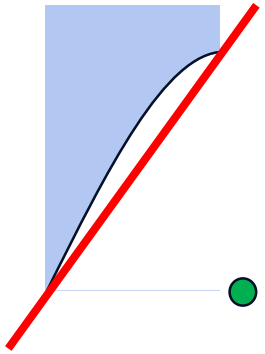
Algorithmic treatment of bilinear constraints

- General form: $a^T z + dx_i x_j \leqq b$ (linear sum plus single product term, inequality or equation)

Consider square case ($i = j$):

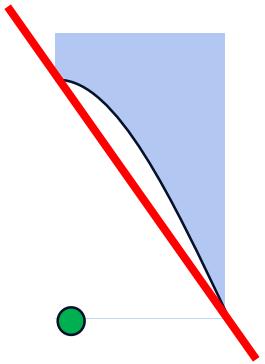
non-convex
$-z - x^2 \leq 0$





branching
$x \leq 0$ or $x \geq 0$

update relaxation locally



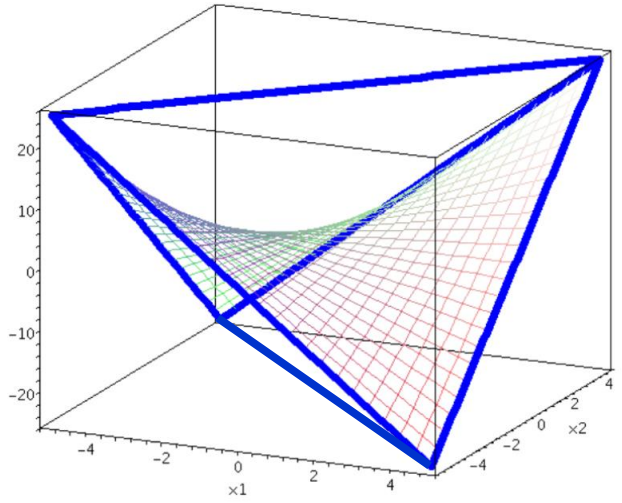
Spatial Branching

Branching variable selection

- What most solvers do: first branching on fractional integer variables as usual
- If no fractional integer variable exists, select continuous variable in violated bilinear constraint
- Our variable selection rule: reduce McCormick volume as much as possible
  - Big McCormick polyhedron is turned into two smaller McCormick polyhedra after branching at LP solution x^*
  - Sum of smaller volumes is smaller than big volume

Branching value selection

- We use a standard way
  - A convex combination of LP value and mid point of current domain



$z_{ij} = x_i\, x_j$

Cutting Planes for Mixed Bilinear Programs

All MILP cutting planes apply

Special cuts for bilinear constraints

- RLT Cuts
  - Reformulation Linearization Technique
  - Multiply linear constraints with single variable, linearize resulting product terms
  - Very powerful for Bilinear Programs, also helps a bit for convex MIQCPs and MILPs
- BQP Cuts
  - Facets from Boolean Quadric Polytope
    - Equivalent to Cut Polytope
  - Currently implemented: triangle inequalities (special case of Padberg's clique cuts for BQP)
  - Helps a bit for Bilinear Programs, convex MIQCPs and MILPs
- PSD Cuts
  - Tangents of PSD cone defined by $Z = xx^T$ relationship: $Z - xx^T \succcurlyeq 0$
  - Not yet implemented in Gurobi

GUROBI OPTIMIZATION

Questions?