Non-Convex Quadratic Optimization with Gurobi

GUROBI OPTIMIZATION

The World's Fastest Solver

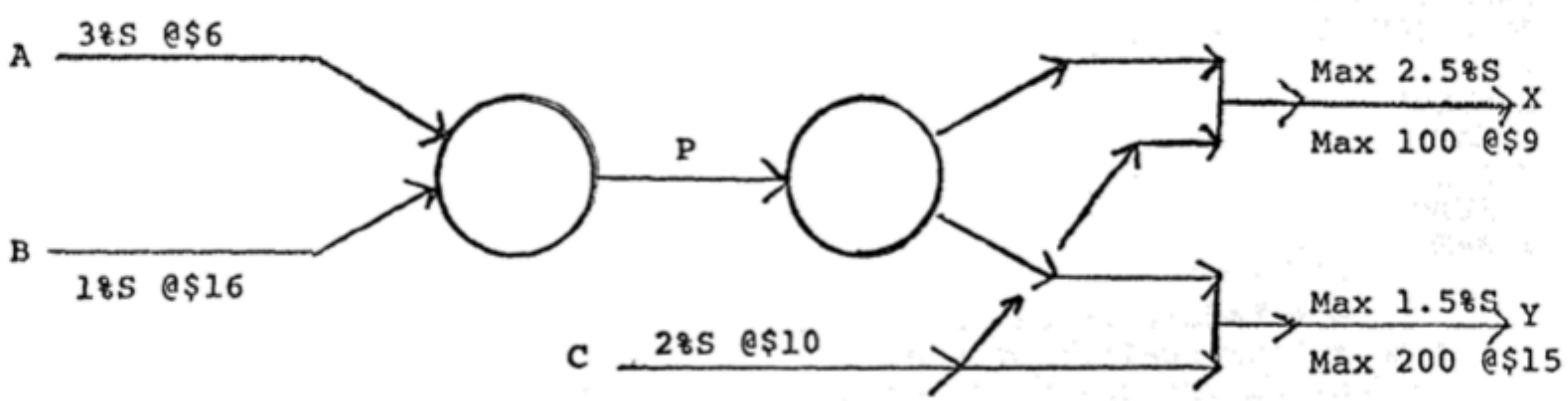
Robert Luce

Motivation: The pooling problem

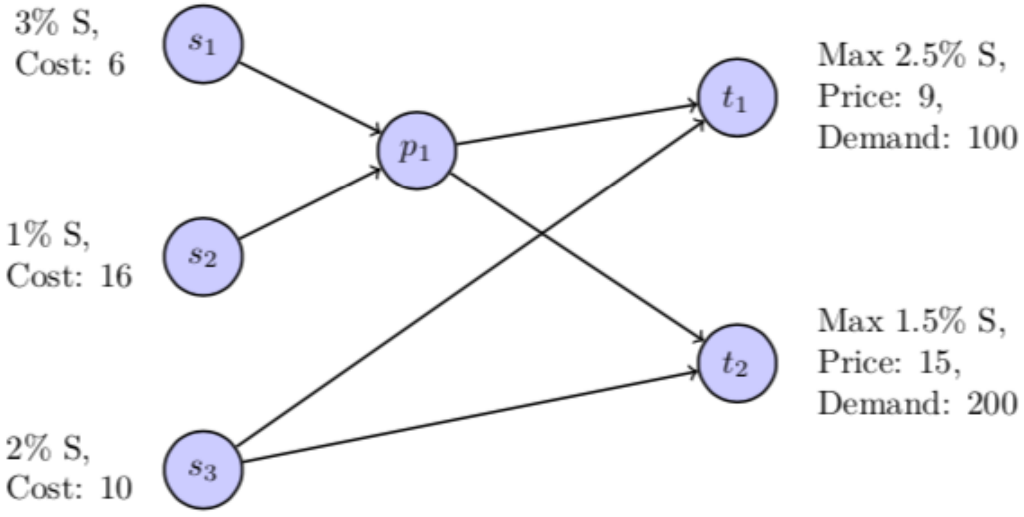
Haverly’s example

Deliver mixtures of different crude oils such that demand and quality is statisfied.

We define a straight forward example as follows:



Using different notation



- Base structure: Minimum cost network flow
  - Arcs (i, j) have a flow f_{ij} with cost & bounds
  - Flow sources s_1, s_2, s_3 and targets t_1, t_2
  - Flow conservation at p_1
- Complication: Flow “quality” (here: pct. sulfur)
  - Each node i has a flow quality variable w_i
  - Sources have fixed quality
  - Sinks have an upper bound on the quality
  - Quality at pool node $i = p_1$ mixes linearly:

$$\sum_{j \in N^-(i)} w_j f_{ji} = w_i \sum_{j \in N^+(i)} f_{ij}$$

Auxiliary notes

Common generalizations

- Multiple pools that mix downstream
- Multiple quality attributes that need to be satisfied at the same time

Other formulations

- The formulation shown goes by the name ”quality formulation”
- Another popular approach: “proportion formulation”
- Or hybrid formulations of the two
- Common to all of them: Quadratic constraints due to quality-of-flow

Consequences of quality-of-flow constraints

Without maintaining quality-of-flow in the network:

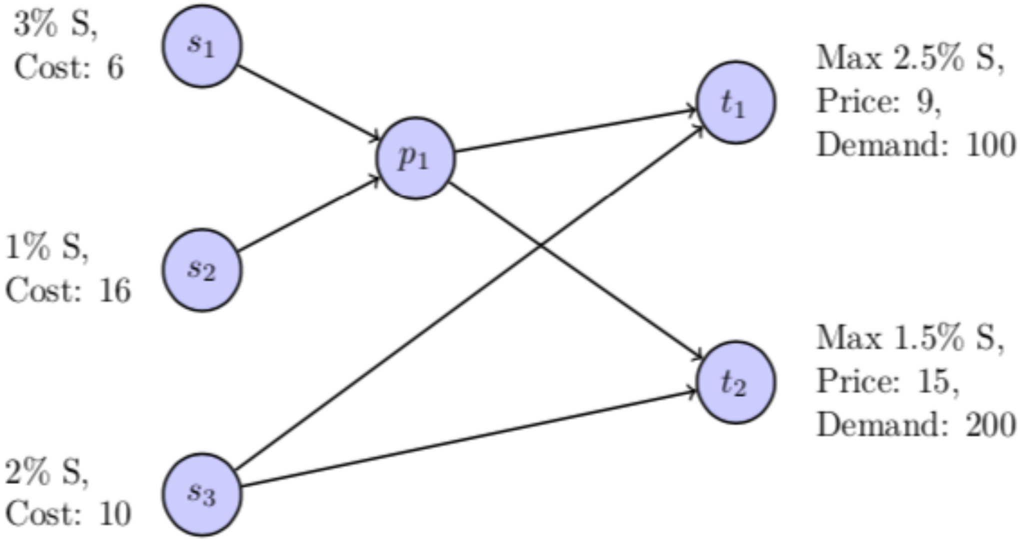
- Pure network flow problem
- Polynomial complexity
- Integer data always results in integer solutions
- Every locally optimal solution is globally optimal

But instead we have:

- Associated decision problem is NP-complete
- Multiple, locally optimal solutions may exist
- Feasible region may have holes, or may even be disconnected

Reason: Quality-of-flow constraint is nonconvex!

What’s nonconvex here



- Quality constraint for pool p_1

$$\sum_{j\,\in N^-(i)} w_j f_{ji} - w_i \sum_{j\,\in N^+(i)} f_{ij} = 0$$

- The feasible set of quadratic equations are typically nonconvex (think of $x^2 = 1$)
- Even the sublevel sets of

$$\sum_{j\,\in N^-(i)} w_j f_{ji} - w_i \sum_{j\,\in N^+(i)} f_{ij}$$

are nonconvex...

Intermezzo: Quadratic functions and convexity

Let $Q \in \mathbb{R}^{n\times n}$ a symmetric matrix, and $q \in \mathbb{R}^n$, and consider the quadratic function

$$f: \mathbb{R}^n \to \mathbb{R},\ x \mapsto x^TQx + q^Tx.$$

Useful properties:

- f is convex iff Q is positive semidefinite
- f is strongly convex iff Q is positive definite
- If Q is positive semidefinite, the sublevel sets $f(x) \le c, c \in \mathbb{R}$, are convex.

Homework:

- What is the matrix representation of the function $f(x, y) = xy$? Is f convex?

A constraint of the form $xy = z$ is sometimes called a *bilinear constraint*. More about that to come!

Nonconvex quadratic optimization with Gurobi

Mixed Integer Quadratically Constrained Programming

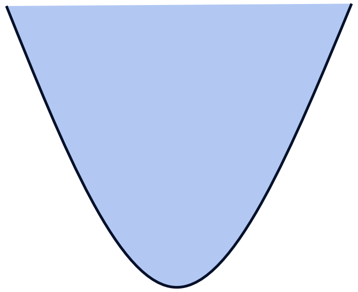
A Mixed Integer Quadratically Constrained Program (MIQCP) is defined as

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

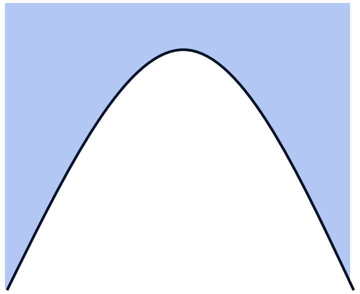
- Q_k are symmetric matrices
- For $Q = Q_k$, any non-zero element $Q_{ij} \neq 0$ gives rise to a product term $Q_{ij} x_i x_j$ in the constraint or objective
- If all Q_k are positive semi-definite, then QCP relaxation is convex
  - MIQCPs with positive semi-definite Q_k can be solved by Gurobi since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

Non-Convex QP, QCP, MIQP, and MIQCP

Prior Gurobi versions: remaining Q constraints and objective after presolve needed to be convex



$x^T Q x \leq b$



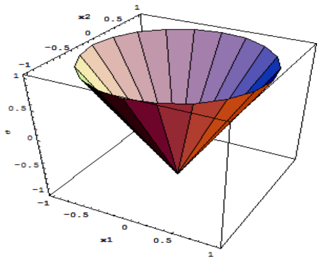
convex
$-z + x^2 \leq 0$

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

If Q is positive semi-definite (PSD) then $x^T Q x \leq b$ is convex

- Q is PSD if and only if $x^T Q x \geq 0$ for all x

But $x^T Q x \leq b$ can also be convex in certain other cases, e.g., second order cones (SOCs)



SOC: $x_1^2 + \cdots + x_n^2 - z^2 \leq 0$

$x^2 + y^2 - z^2 \leq 0, z \geq 0$: at level z, (x, y) is a disc with radius z

Non-Convex QP, QCP, MIQP, and MIQCP

Prior Gurobi versions could deal with two types of non-convexity

- Integer variables
- SOS constraints

Gurobi 9.0 can deal with a third type of non-convexity

- Bilinear constraints

All these non-convexities are treated by

- Cutting planes
- Branching

Translation of non-convex quadratic constraints into bilinear constraints

$3x_1^2 - 7x_1x_2 + 2x_1x_3 - x_2^2 + 3x_2x_3 - 5x_3^2 = 12$ (non-convex Q constraint)

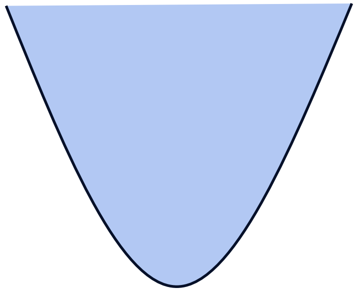
$z_{11} \coloneqq x_1^2, z_{12} \coloneqq x_1x_2, z_{13} \coloneqq x_1x_3, z_{22} \coloneqq x_2^2, z_{23} \coloneqq x_2x_3, z_{33} \coloneqq x_3^2$ (6 bilinear constraints)

$\Rightarrow$ $3z_{11} - 7z_{12} + 2z_{13} - z_{22} + 3z_{23} - 5z_{33} = 12$ (linear constraint)

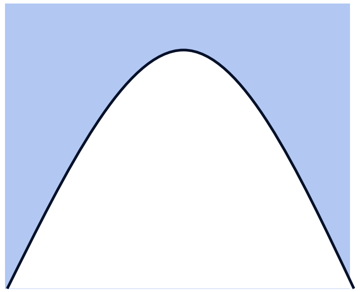
Dealing With Bilinear Constraints

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

Consider square case ($x = y$):



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

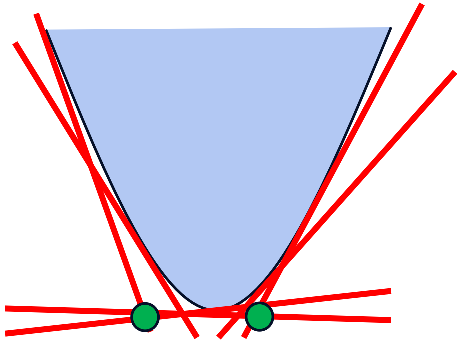


non-convex
$-z - x^2 \leq 0$

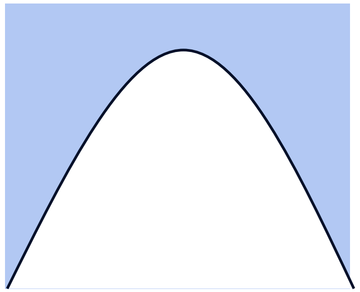
Dealing With Bilinear Constraints

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

Consider square case ($x = y$):



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



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

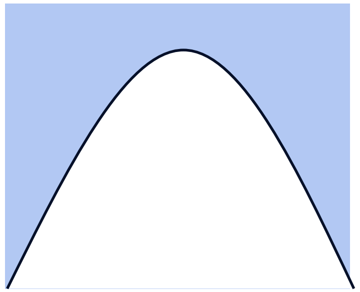
easy: add tangent cuts

Dealing With Bilinear Constraints

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

Consider square case ($x = y$):

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

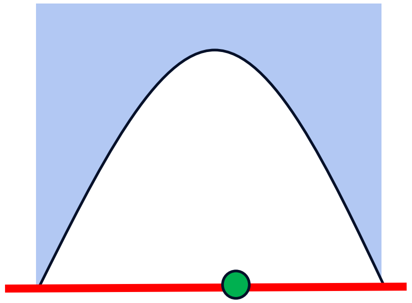


Dealing With Bilinear Constraints

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

Consider square case ($x = y$):

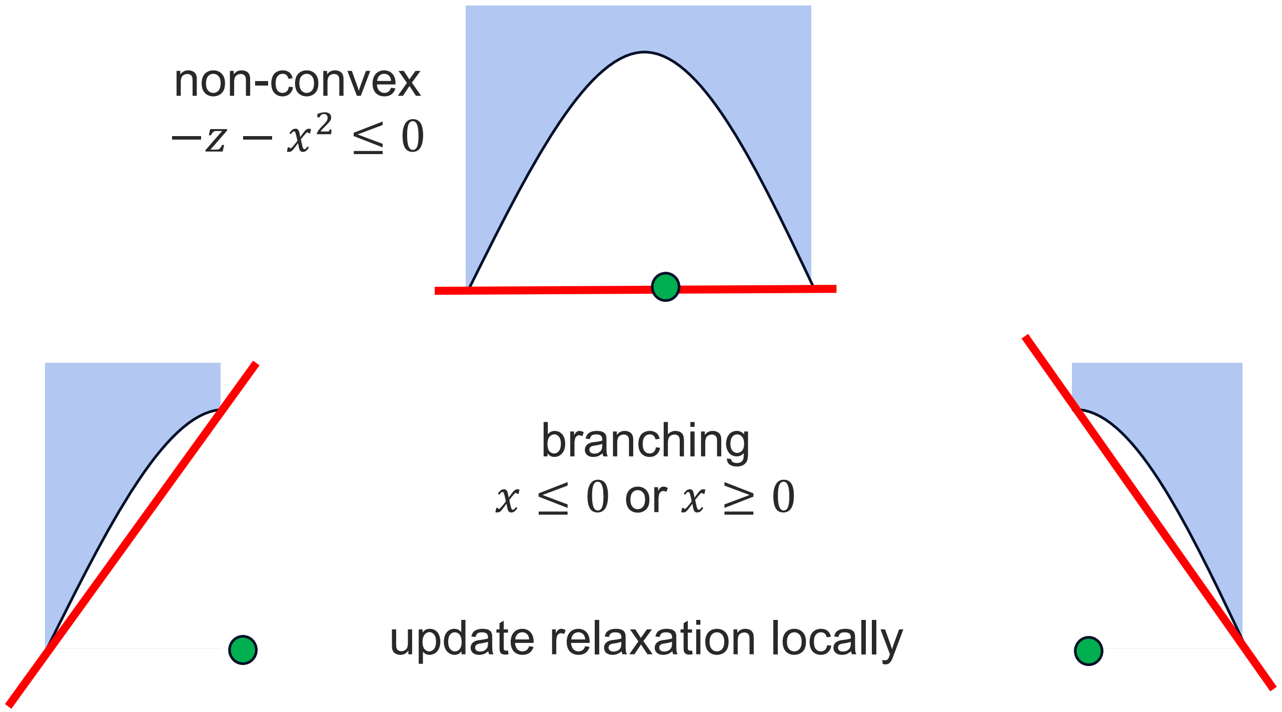
non-convex
$-z - x^2 \leq 0$



Dealing With Bilinear Constraints

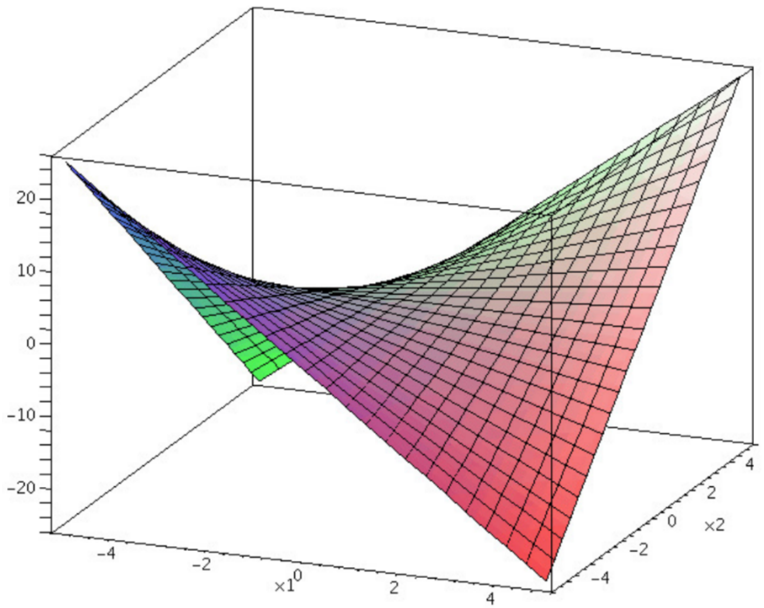
General form: $a^T z + dxy \leqq b$ (linear sum plus single product term, inequality or equation)

Consider square case ($x = y$):



LP Relaxation of Bilinear Constraints

Mixed product case: $-z + xy = 0$

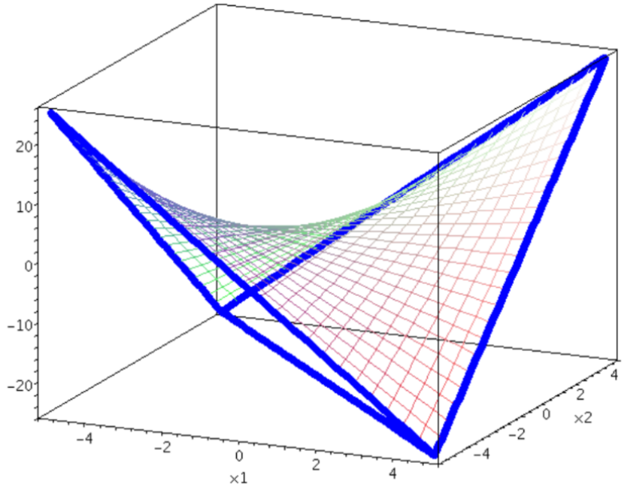


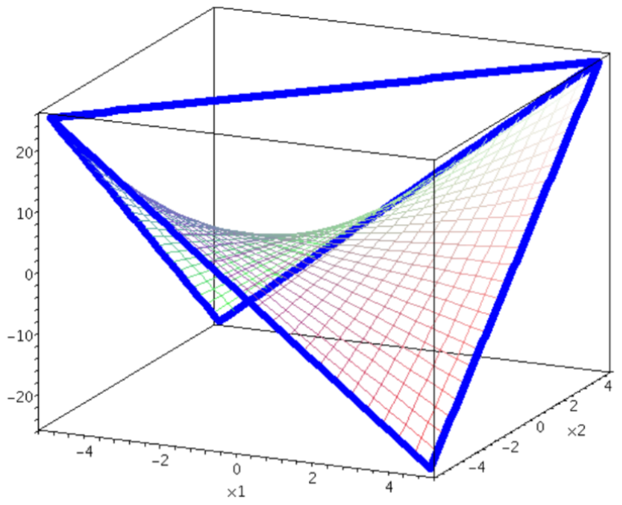
McCormick lower and upper envelopes:

$$-z + l_x y + l_y x \leq l_x l_y$$
$$-z + u_x y + u_y x \leq u_x u_y$$

$$-z + u_x y + l_y x \geq u_x l_y$$
$$-z + l_x y + u_y x \geq l_x u_y$$

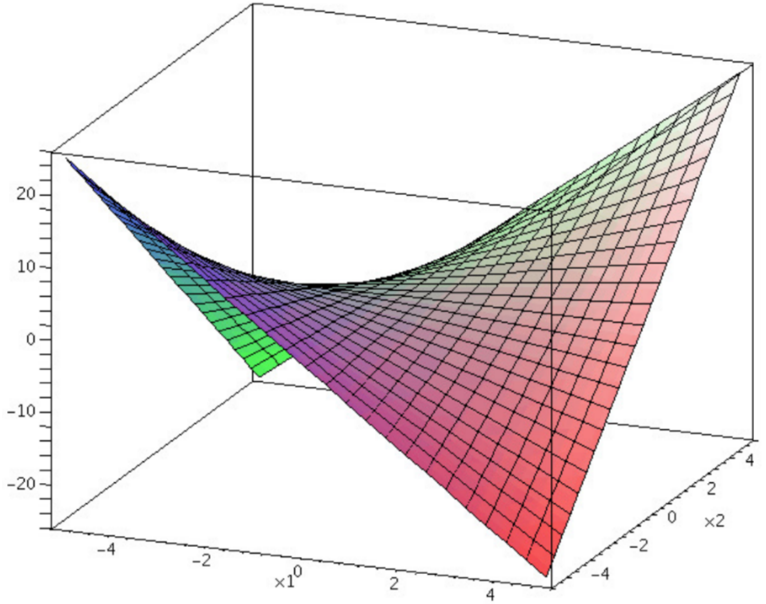
pictures from Costa and Liberti: "Relaxations of multilinear convex envelopes: dual is better than primal"





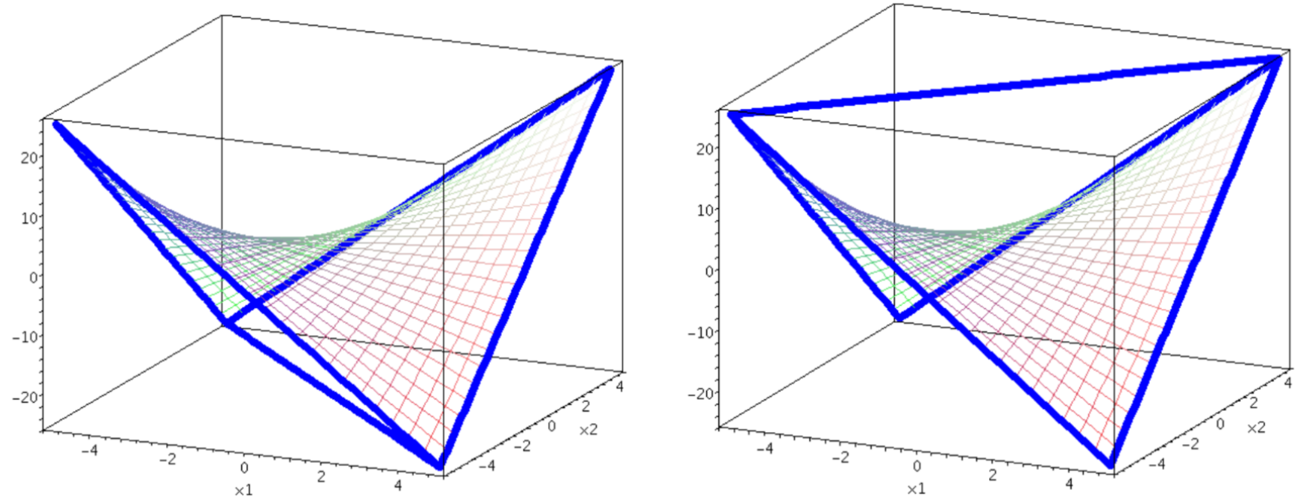
LP Relaxation of Bilinear Constraints

Mixed product case: $-z + xy = 0$



McCormick lower and upper envelopes:

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



$$-z + l_x y + l_y x \le l_x l_y$$
$$-z + u_x y + u_y x \le u_x u_y$$

$$-z + u_x y + l_y x \ge u_x l_y$$
$$-z + l_x y + u_y x \ge l_x u_y$$

coefficients depend on local bounds

Adaptive Constraints in LP Relaxation

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

Alternative to adaptive constraints: locally valid cuts

- Add tighter McCormick relaxation on top of weaker, more global one, to local node
- Advantages:
  - old simplex basis stays valid in all cases
    - more global McCormick constraints will likely become slack and basic
  - should lead to fewer simplex iterations
- Disadvantages:
  - basis size (number of rows) changes all the time during solve
    - refactorization needed
    - complicated (and potentially time and memory consuming) data management needed
  - redundant more global McCormick constraints stay in LP
    - LP solver performs useless calculations in linear system solves

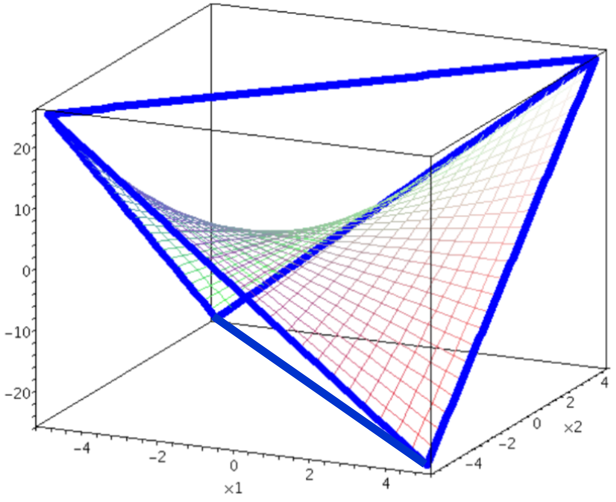
Spatial Branching

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 is a combination of:
  - sum of absolute bilinear constraint violations
  - 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
  - shadow costs of variable for linear constraints

Branching value selection

- We use a standard way
  - a convex combination of LP value and mid point of current domain
- Avoid numerical pitfalls
  - large branching values for unbounded variables
  - tiny child domains if LP value is very close to bound
  - very deep dives (node selection)



$z_{ij} = x_i \, x_j$

Cutting Planes for Mixed Bilinear Programs

All MILP cutting planes apply

Special cuts for bilinear constraints

- RLT Cuts
  - Reformulation Linearization Technique (Sherali and Adams, 1990)
  - 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 (Padberg 1989)
    - equivalent to Cut Polytope
  - currently implemented: triangle inequalities (special case of Padberg's clique cuts for BQP)
- PSD Cuts
  - tangents of PSD cone defined by $Z = xx^T$ relationship: $Z - xx^T \succcurlyeq 0$ (Sherali and Fraticelli, 2002)
  - not yet implemented in Gurobi

Thank You!

GUROBI OPTIMIZATION

The World's Fastest Solver