Non-Convex Quadratic Optimization

Gurobi 9.0

GUROBI OPTIMIZATION

The World's Fastest Solver

Speaker Introduction

Dr. Tobias Achterberg

- Director of R&D at Gurobi Optimization
- Formerly a developer at ILOG, where he worked on CPLEX 11.0 to 12.6
- Obtained his degree in mathematics and computer science from the Technical University of Berlin and the Zuse Institute Berlin, then finished doctorate in mathematics with Prof. Martin Grötschel in 2007
- Dr. Achterberg is the author of SCIP which is regarded as the best academic MIP solver

Speaker Introduction

Dr. Eli Towle

- Optimization Support Engineer at Gurobi
- Dr. Eli Towle has a PhD in Industrial and Systems Engineering from the University of Wisconsin – Madison
- His research focused on stochastic network interdiction models and polyhedral relaxations of certain nonconvex sets

Mixed Integer Quadratically Constrained Programming

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

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

- 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

Applications

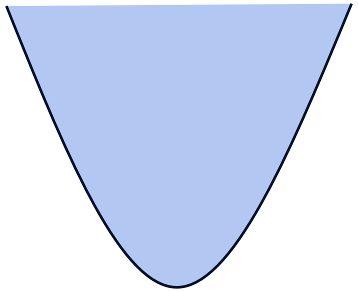
- Pooling problem (blending problem is LP, pooling introduces intermediate pools $\rightarrow$ bilinear)
- Petrochemical industry (oil refinery: constraints on ratio of components in tanks)
- Wastewater treatment
- Emissions regulation
- Agricultural / food industry (blending based on pre-mix products)
- Mining
- Energy
- Production planning (constraints on ratio between internal and external workforce)
- Logistics (restrictions from free trade agreements)
- Water distribution (Darcy-Weisbach equation for volumetric flow)
- Engineering design
- Finance (constraints on exchange rates)

General MINLP

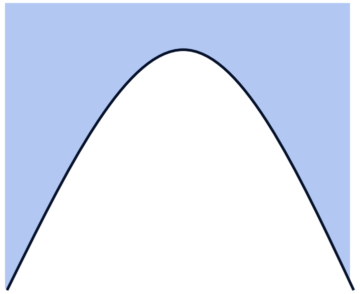
- Non-convex MIQCP solves (in theory) polynomial problems of arbitrary degree
- Solve general MINLPs by approximating as polynomial problem
  - but: will often fail for higher degrees due to numerical issues

Non-Convex QP, QCP, MIQP, and MIQCP

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



$x^TQx \le b$



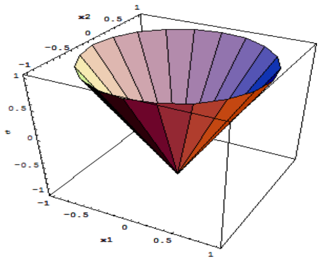
convex
$-z + x^2 \le 0$

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

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

- Q is PSD if and only if $x^TQx \ge 0$ for all x

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



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

$x^2 + y^2 - z^2 \le 0, z \ge 0$: 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

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)**

$$\Rightarrow \quad \mathrm{z}_{11} := x_1^2, \mathrm{z}_{12} := x_1x_2, \mathrm{z}_{13} := x_1x_3, \mathrm{z}_{22} := x_2^2, \mathrm{z}_{23} := x_2x_3, \mathrm{z}_{33} := x_3^2$$ **(6 bilinear constraints)**

$$3\mathrm{z}_{11} - 7\mathrm{z}_{12} + 2\mathrm{z}_{13} - \mathrm{z}_{22} + 3\mathrm{z}_{23} - 5\mathrm{z}_{33} = 12$$ **(linear constraint)**

More Details on Bilinear Transformation

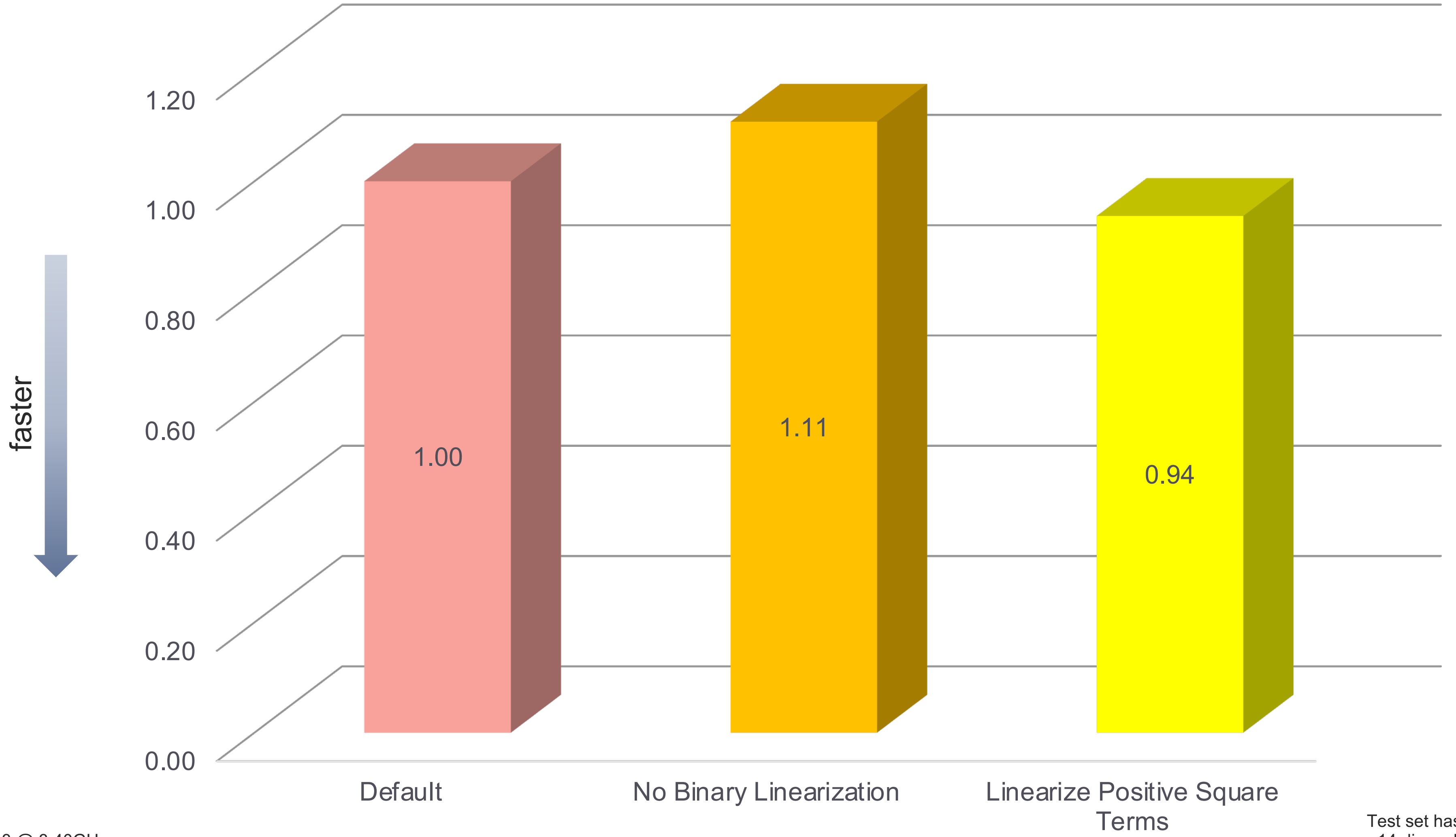
For each term $a_{ij}x_ix_j$ in a non-convex quadratic constraint:

- If x_i and/or x_j are fixed, move to linear part or right hand side of constraint;
- Else if $i = j$ and x_i is binary, replace x_i^2 by x_i and move term to linear part of constraint;
- Else if x_i or x_j is binary, introduce $z_{ij} \coloneqq x_ix_j$, move $a_{ij}x_ix_j = a_{ij}z_{ij}$ to linear part, and
  - if possible, add big-M linearization for $z_{ij} \coloneqq x_ix_j$
  - otherwise, add SOS1 formulation for $z_{ij} \coloneqq x_ix_j$;
- Else if $i = j$, $a_{ij} > 0$, and the Q constraint is a $\leq$ inequality, keep term in quadratic part;
- Else: introduce $z_{ij} \coloneqq x_ix_j$, move $a_{ij}x_ix_j = a_{ij}z_{ij}$ to linear part, and add the bilinear constraint
  - $z_{ij} = x_ix_j$, if the Q constraint is an equation;
  - $z_{ij} \geq x_ix_j$, if the Q constraint is a $\leq$ inequality, and $a_{ij} > 0$;
  - $z_{ij} \leq x_ix_j$, if the Q constraint is a $\leq$ inequality, and $a_{ij} < 0$.

More sophisticated partitions into convex and non-convex parts are possible and may work better!

Performance Impact of Bilinear Translation

GUROBI OPTIMIZATION



Timeouts:	0	14	0

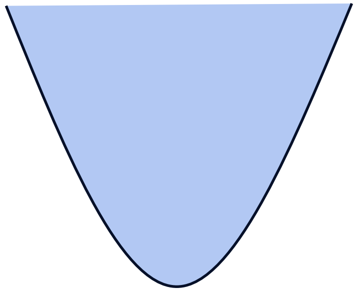
Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v3 @ 3.40GHz
4 cores, 8 hyper-threads
32 GB RAM

Test set has 444 models, using 5 random seeds:
- 14 discarded due to inconsistent answers
- 54 discarded that none of the versions can solve
- speed-up measured on >10s bracket: 116 models

Dealing With Bilinear Constraints

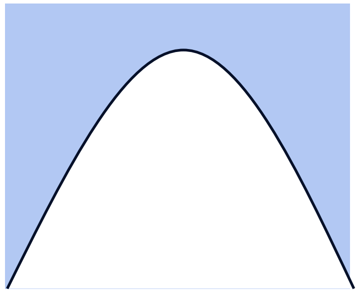
General form: $a^T z + dxy \leqq b$ (linear sum plus single product term, inequality or equation)

Consider square case ($x = y$):



convex

$-z + x^2 \leq 0$



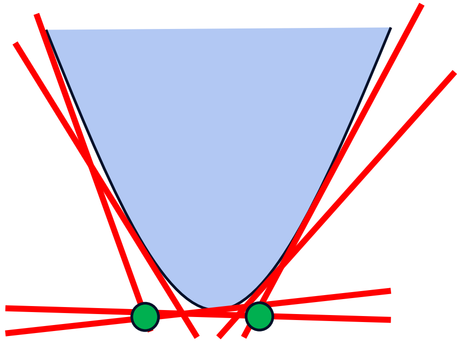
non-convex

$-z - x^2 \leq 0$

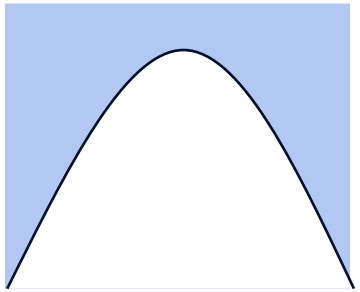
Dealing With Bilinear Constraints

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
$-z - x^2 \leq 0$

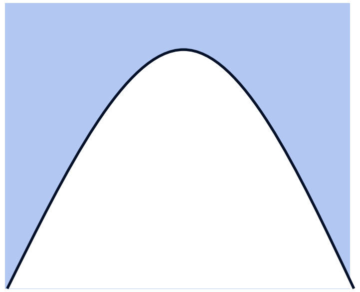
easy: add tangent cuts

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
$-z - x^2 \leq 0$

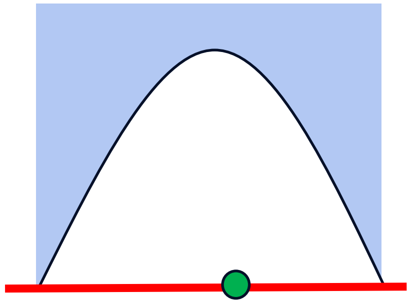


Dealing With Bilinear Constraints

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

Consider square case ($x = y$):

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

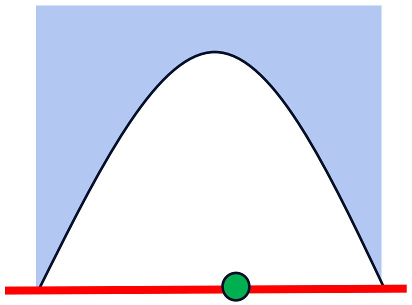


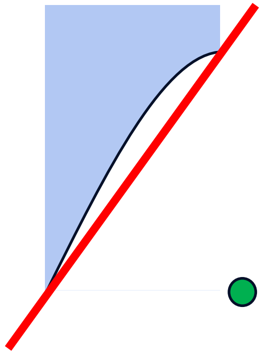
Dealing With Bilinear Constraints

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

Consider square case ($x = y$):

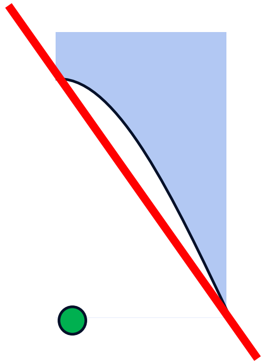
non-convex
$-z - x^2 \leq 0$





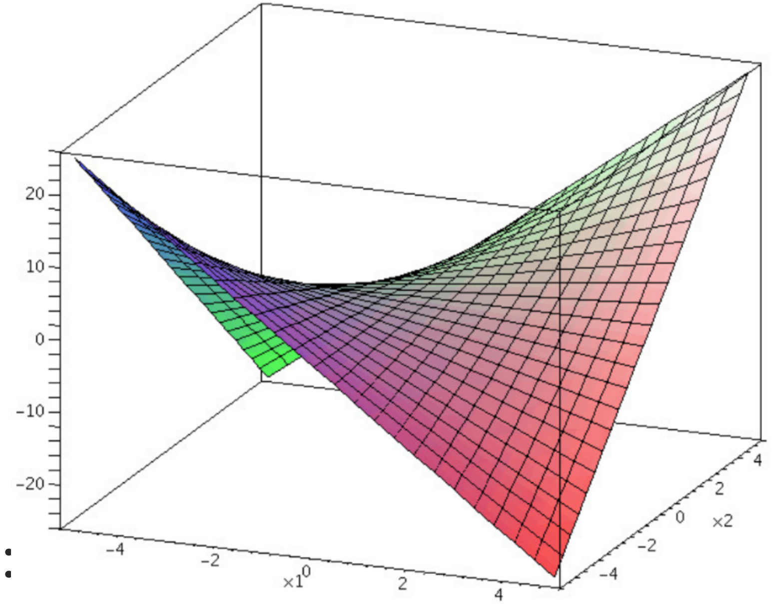
branching
$x \leq 0$ or $x \geq 0$

update relaxation locally



LP Relaxation of Bilinear Constraints

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

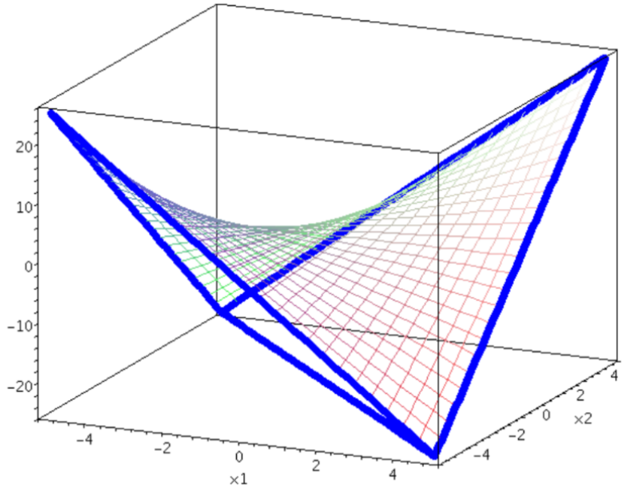


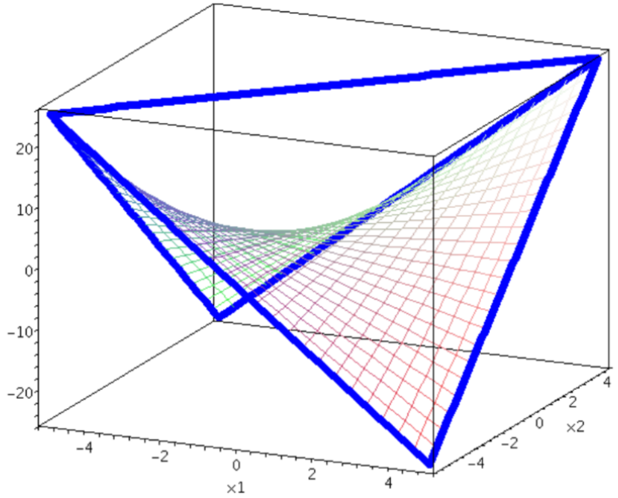
McCormick lower and upper envelopes:

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

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

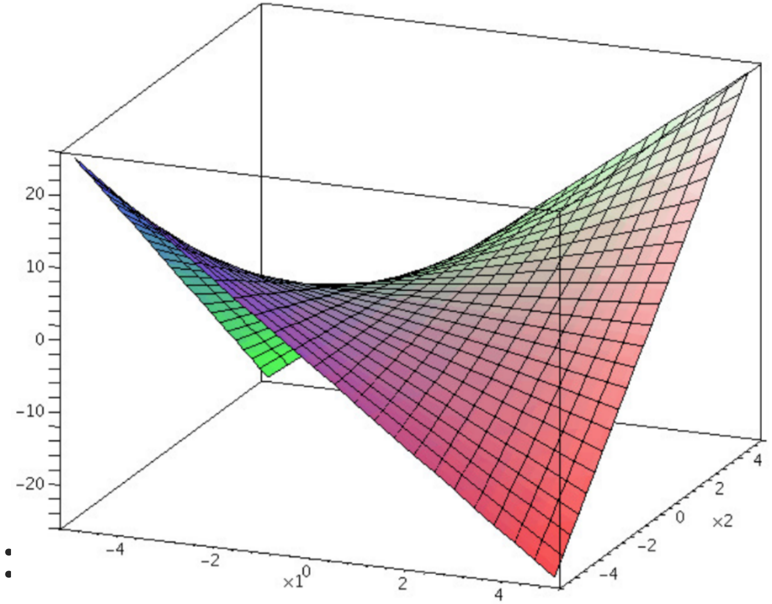
pictures from Costa and Liberti: "Relaxations of multilinear convex envelopes: dual is better than primal"





LP Relaxation of Bilinear Constraints

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



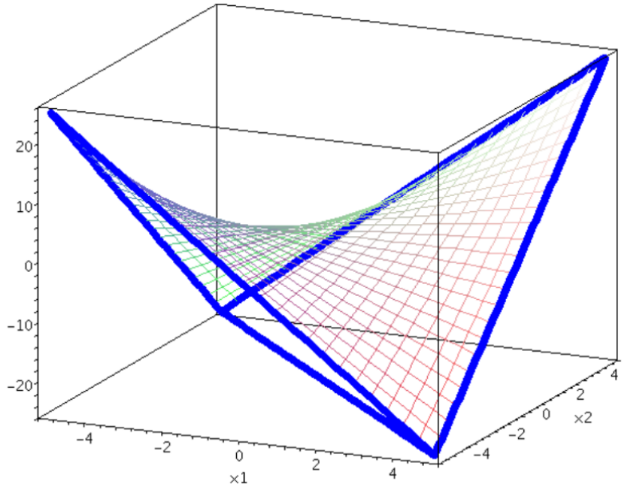
McCormick lower and upper envelopes:

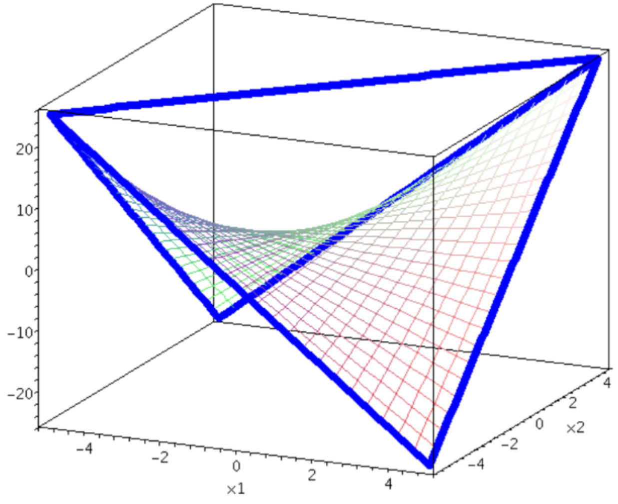
$$-z + l_x y + l_y x \leq l_x l_y$$
$$-z + u_x y + u_y x \leq u_x u_y$$

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

coefficients depend on local bounds

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





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

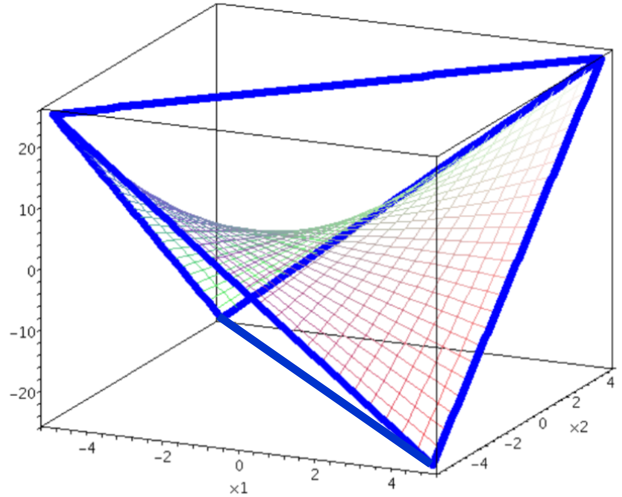
Spatial Branching

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

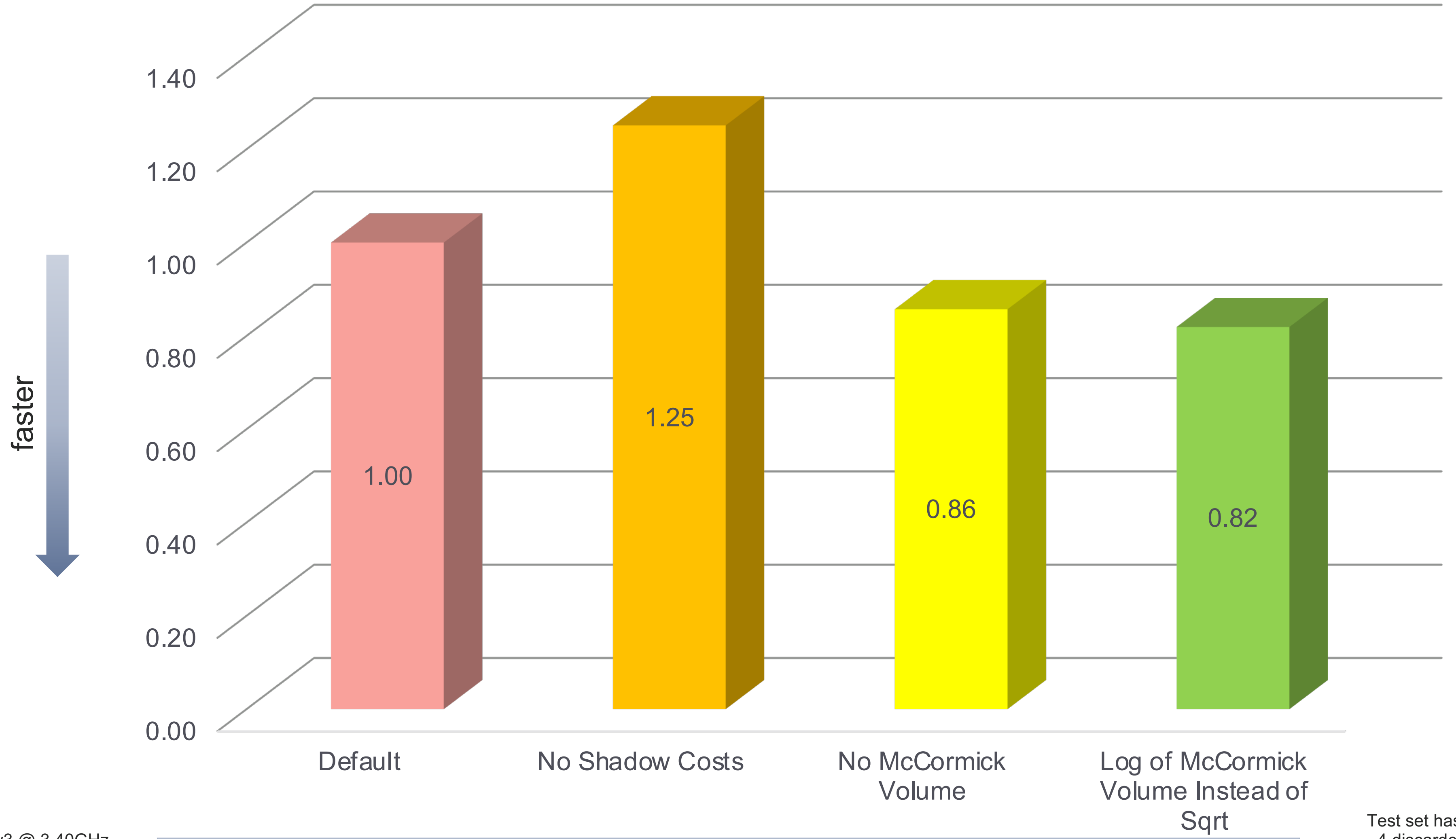
Branching value selection

- 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)



$z_{ij} = x_i \, x_j$

Performance Impact of Branching



Timeouts:	10	60	8	5

Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v3 @ 3.40GHz
4 cores, 8 hyper-threads
32 GB RAM

Test set has 444 models, using 5 random seeds:
- 4 discarded due to inconsistent answers
- 54 discarded that none of the versions can solve
- speed-up measured on >10s bracket: 143 models

Cutting Planes for Mixed Bilinear Programs

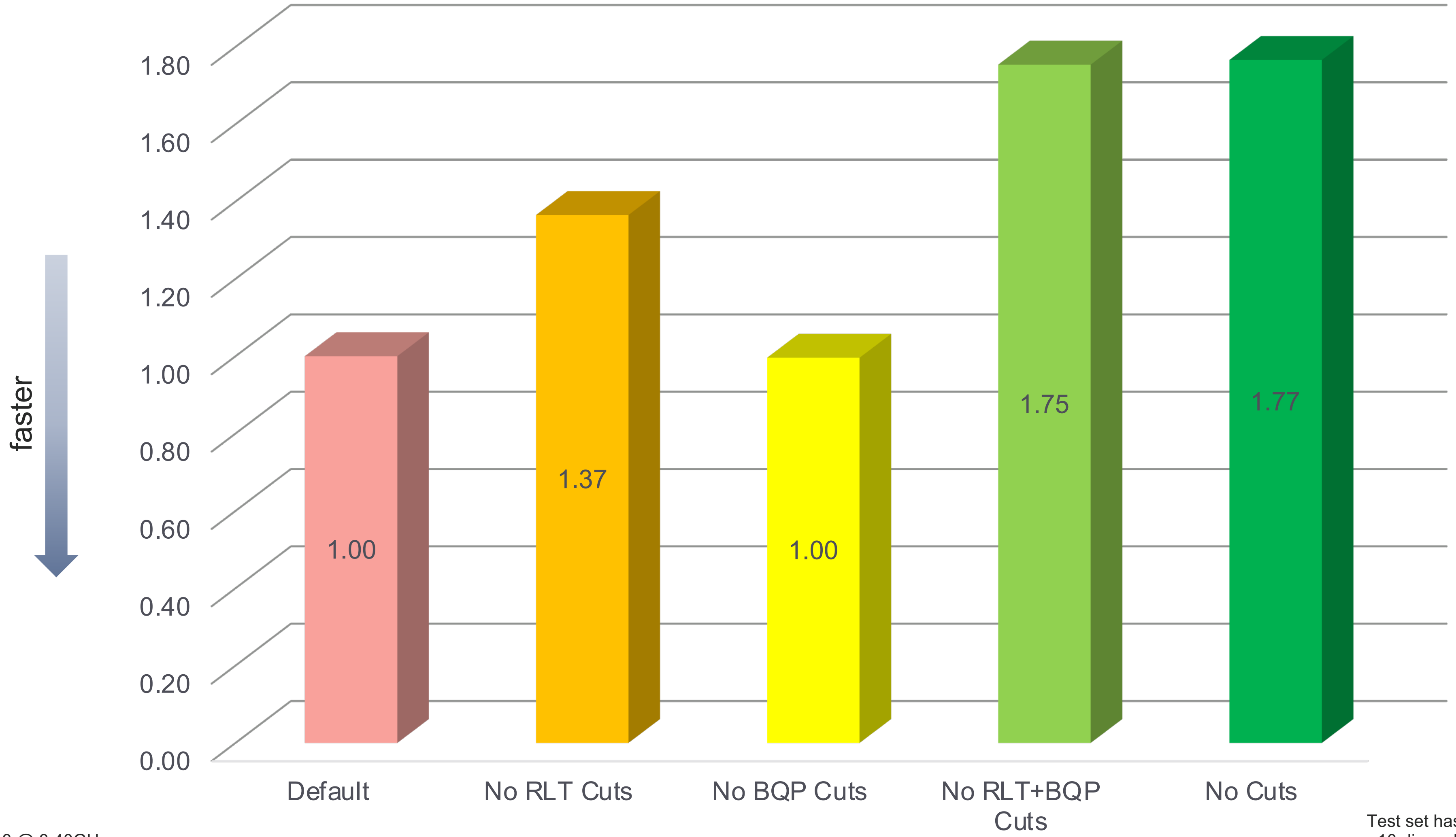
GUROBI OPTIMIZATION

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

Performance Impact of Cutting Planes



Timeouts:	20	110	20	165	187

Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v3 @ 3.40GHz
4 cores, 8 hyper-threads
32 GB RAM

Test set has 444 models, using 5 random seeds:
- 10 discarded due to inconsistent answers
- 53 discarded that none of the versions can solve
- speed-up measured on >10s bracket: 152 models

Thank You – Questions?

GUROBI OPTIMIZATION

The World's Fastest Solver

Your Next Steps

- **Try Gurobi 9.0 Now!**
  - Get a 30-day commercial trial license of Gurobi at www.gurobi.com/free-trial
  - Academic and research licenses are free!
- **For questions about Gurobi pricing, please contact sales@gurobi.com or sales@gurobi.de**
- **A recording of this webinar, including the slides, will be available in about one week**

- **Upcoming webinars with more details on individual features**
  - January 28 and 29: How to Choose a Math Solver
  - February: Compute Server and Cluster Manager
  - See www.gurobi.com/events