Gurobi 9.1 Performance

GUROBI OPTIMIZATION

The World's Fastest Solver

Zonghao Gu
December 2020

Performance Improvements

- **LP improvements**
  - Dual: 29% faster overall, 66% faster for > 100s models
  - Primal: 17% faster overall, 37% faster for > 100s models
  - Barrier: 15% faster overall, 34% faster for > 100s models
- **MIP improvements**
  - MILP: 5% faster overall, 10% faster for > 100s models
  - MIQP (convex): 6% faster overall, 20% faster for > 100s models
  - MIQCP (convex): 13% faster overall, 57% faster for > 100s models
- **Bilinear or nonconvex MIQCP improvements**
  - 4.1x overall, 9.6x for > 100s models
- **IIS improvements**
  - 2.6x overall, 5.7x for >100s models

LP Performance

LP Improvements

- Presolve
  - Improved a presolve reduction
  - Improved dependent row reduction
- Better decision to solve dual formulation
  - Use machine learning to decide
  - Including deciding which method to use, primal or dual
- Weak symmetry improvement

LP Improvements, Simplex

- **Dual pricing strategy**
  - Improve handling among devex, different types of steepest edge pricing
- **Scaling, especially objective scaling**
- **Perturbation**
- **LU factorization**
  - 2x2 block pivoting
  - Improvement of sparse vs dense treatment
  - Pivoting candidate selection

LP Improvements, Barrier

- **Crossover**
  - Improved ratio test for primal pushes
  - Better numeric handling
    - Initial crossover basis
    - Etc.
- **Barrier parallel improvement**
  - Especially for machines with more than four physical cores

Approaches to Solve Dual Formulation

- **Our approach**
  - Use the original model to formulate the dual model
  - Apply presolve on the dual model
  - Solve the presolved model
- **Alternative approach**
  - Apply presolve to the original model to get the presolved model
  - Formulate the dual model based on the presolved model
  - Solve the dual of the presolved model

Decision to Solve the Dual Formulation

- **Estimate the size of the dual better**
  - Exclude rows and columns in the dual formulation that will obviously be removed by presolve
- **Use machine learning**
  - Find important factors to decide whether to solve the dual formulation
    - We fed the data to scikit-learn – it identified key inputs
      - The aspect ratio, # columns divided by # rows
      - Similar to what we were doing before
  - Decide which method, primal or dual, to solve dual formulation
    - ML gave us a nice formula to decide
      - Mostly expected or understandable, but not all
      - We manually adjusted a bit

Weak Symmetry for LP (aka LP Folding)

- **Example**
  - $3w + 3x + 3y + 3z \leq 11$
  - $3w + 3x + 2y + 4z \leq 11$
  - $3w + 3x + 4y + 2z \leq 11$

  - x, y don't look symmetric, but
    - Sum of coefficients = 9 for each column
    - Sum of coefficients = 12 for each row
- **The conditions for weak symmetry**
  - Divide the rows and columns into classes
  - The sum of the coefficients in a row is equal for each row in the same class
  - The sum of the coefficients in a column is equal for each column in the same class
  - Objective coefficients and bounds are the same for the variables in the same class
  - Rhs and senses are the same for the rows in the same class

Property of LP Weak Symmetry

- **Given a solution x^***
  - Let $x'_j = \frac{1}{n}\sum_{i \in C} x^*_i, |C| = n, \forall j \in C$, for any variable class C
  - Easy to show x' is also a feasible solution with the same objective value
    - We can let all the variables in the same class equal
    - All the rows in the same class will be the identical
- **Symmetry reduced model**
  - Combine all the variables in a variable class together
    - by adding up coefficients in the rows
  - Keep only one row for each row class
- **References**
  - Several reports with computational results
    - Many LP solvers have the feature
  - We have it since Gurobi 7.5
  - The key part is to convert nonbasic symmetric solution to basic one

Weak Symmetry Improvement

- **Detection**
  - Catch more general case, including the example in slide 9
  - Speedup: handle sparsity and hashing better
- **Converting to basic solution**
  - Simplex
    - 9.1 uses crossover to convert nonbasic solution to basic one
      - Initial crash basis construction
      - Check many different numeric bad signs, restart if bad enough
      - Heavily tested and refined
    - 9.0 uses the superbasic code to convert
      - Only used for corner cases
  - Barrier
    - 9.1 does crossover twice
      - First crossover on the smaller model is cheap
      - Second crossover with clean solution is numerically more stable
    - 9.0 does crossover once
      - Uncrush the barrier solution (not very clean) to the solution for the large model
      - Crossover with not clean solution on the large model

MIP Performance

Heuristics

- **New heuristics**
  - NoRel heuristic
  - Some new variants of RINS
- **Improvements of existing heuristics**
  - Adjustment on SubMIP heuristic setting
  - Adjustment on Improvement heuristic
- **Performance**
  - Improved MIP performance (optimality) by 1% to 2% overall
  - Greatly improved performance for finding better solutions

Results for Finding Better solutions

- **Test set**
  - All MIPLIB 2017 open problems: 245 models
- **Runs**
  - One hour run with 9.0 default, 9.1 default and 9.1 NoRel heuristic
- **Winning measure:**
  - Solution is at least 1% better in terms of the objective value
    - If one run doesn't find any feasible solution in an hour, then the run finding a feasible solution is considered as winner
- **9.0 default vs 9.1 default**
  - 85 models with the solution difference by more than 1%
    - 16 wins for 9.0 vs 69 wins for 9.1
- **9.1 default vs 9.1 NoRel heuristic**
  - 119 models with the solution difference by more than 1%
    - 29 wins for 9.1 vs 90 wins for 9.1 NoRel

Outer Approximation, Tangent Cut for MIQCP

- **Outer approximation method to solve MIQCP**
  - Solve LP relaxation
  - Add tangent cuts for quadratic constraints to LP relaxation
- **Forms of quadratic constraints**
  - Standard form, SOC (second order cone)
    - $\sum x_i^2 \leq y^2$
    - It often needs to add new variables and to do L'L factorization
  - General form
    - $\sum q_{ij}\, x_i\, x_j + \sum a_j\, x_j \leq b$
  - Input
    - General form, which covers SOC
  - Internal
    - Controlled by parameter PreMIQCPForm
      - -1 auto
      - 0 general form
      - 1 SOC
      - 2 disaggregated SOC

Tangent Cut for Quadratic Constraint

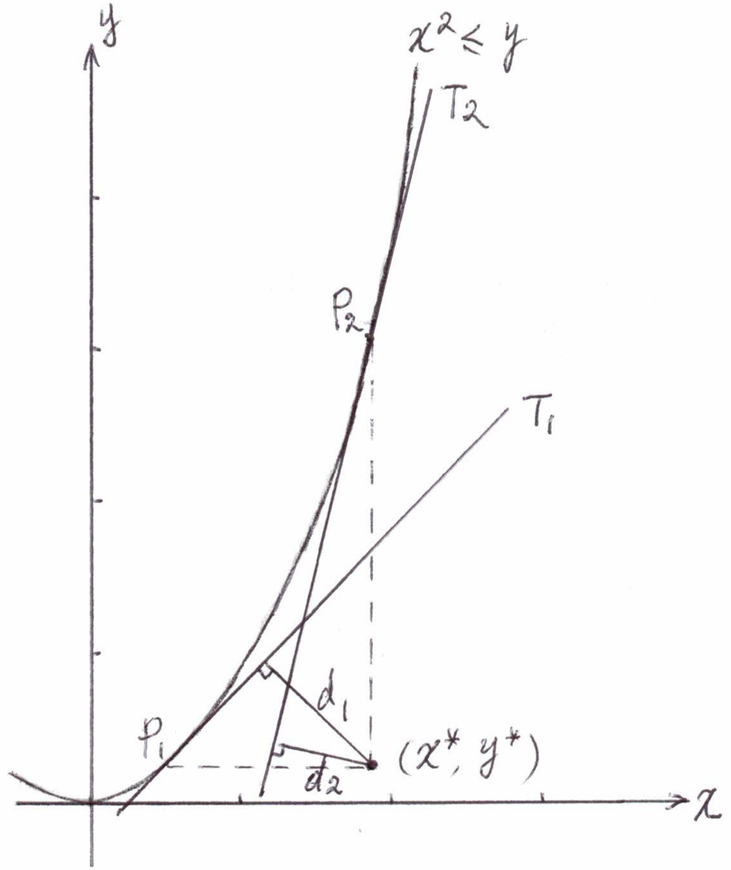
- **Many options for cutting off LP relaxation solution x^***
  - Which tangent plane is best?
    - Best = maximum violation?
    - Best = quick separation?

Tangent Cut for SOC

- For given relaxation solution (x^*, y^*) with $\sum x_i^{*2} > y^{*2}$
  - i.e. (x^*, y^*) violates $\sum x_i^2 \leq y^2$
- Let $\mathrm{y}' = \sqrt{\sum x_i^{*2}}$ i.e. find a point (x^*, y') on SOC surface
- Use point (x^*, y') to compute the tangent plane
  - It cuts off (x^*, y^*)
  - The distance to (x^*, y^*) is maximum among all the tangent planes cutting off (x^*, y^*)

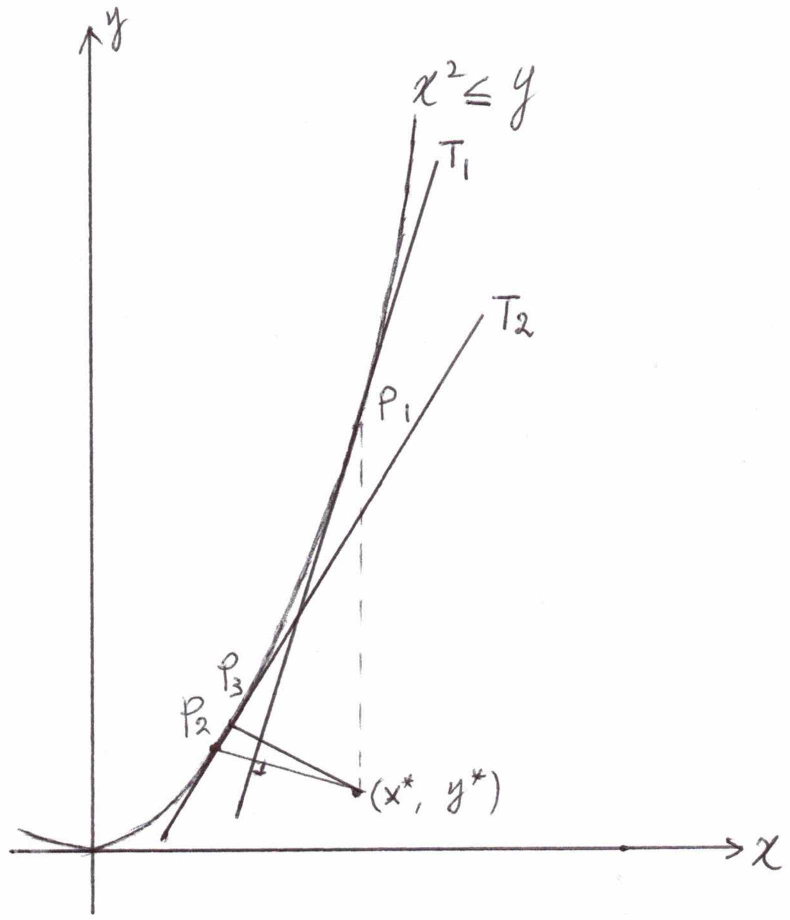
Tangent Cut for Quadratic Constraints in General Form

- **How to find a point on surface**
  - For given x^* with
    $\sum q_{ij}\, x_i^* x_j^* + \sum a_j\, x_j^* > \ b$
  - It isn't easy to find a point on surface with the tangent plane cutting off x^*
  - There are many ways to find such a point, example $x^2 \leq y$
    - Violated point P(x^*, y^*)
    - Keep y^* unchanged to project to the surface, tangent cut T_1 with distance d_1
    - Keep x^* unchanged to project to the surface, tangent cut T_2 with distance d_2
    - Which tangent plane is better?



Finding Tangent Cut with Maximum Distance

- **Our iterative approach**
  - Given a violated point (x^*, y^*)
  - Find a reasonably good point P_1 on surface, whose tangent plane cuts off (x^*, y^*), call tangent plane T_1
  - Project (x^*, y^*) to T_1 and extend it to P_2, then generate tangent plane T_2
  - Project (x^*, y^*) to T_2 …
  - Until P_n is very close to the projection of (x^*, y^*) to T_{n-1}
  - Use P_n to generate the tangent cut T_n



Performance Impact of Q Tangent Cut Improvement

- Internal convex MIQCP set
  - 3.3% overall, 10% for > 100s models
- Internal nonconvex MIQCP set
  - 1% overall

Gurobi 9.1 – Performance Summary

GUROBI OPTIMIZATION

- Performance improvements compared to Gurobi 9.0

Algorithm	Overall speed-up	On >100sec models
Primal simplex	17%	37%
Dual simplex	29%	66%
Barrier	15%	34%
MILP	5%	10%
Convex MIQP	6%	20%
Convex MIQCP	13%	57%
Non-convex MIQCP	4.1x	9.6x
IIS detection	2.6x	5.7x