Presolve and Cutting Planes

Presolve and Cutting Planes

"Tighter" formulations

- Original MIP formulation can almost always be improved
  - Fewer constraints and variables
    - Less data to process
  - Smaller difference between space of feasible continuous and feasible integer solutions
- Two techniques:
  - Presolve and cutting planes

Model Reformulation

“Tighten” formulation

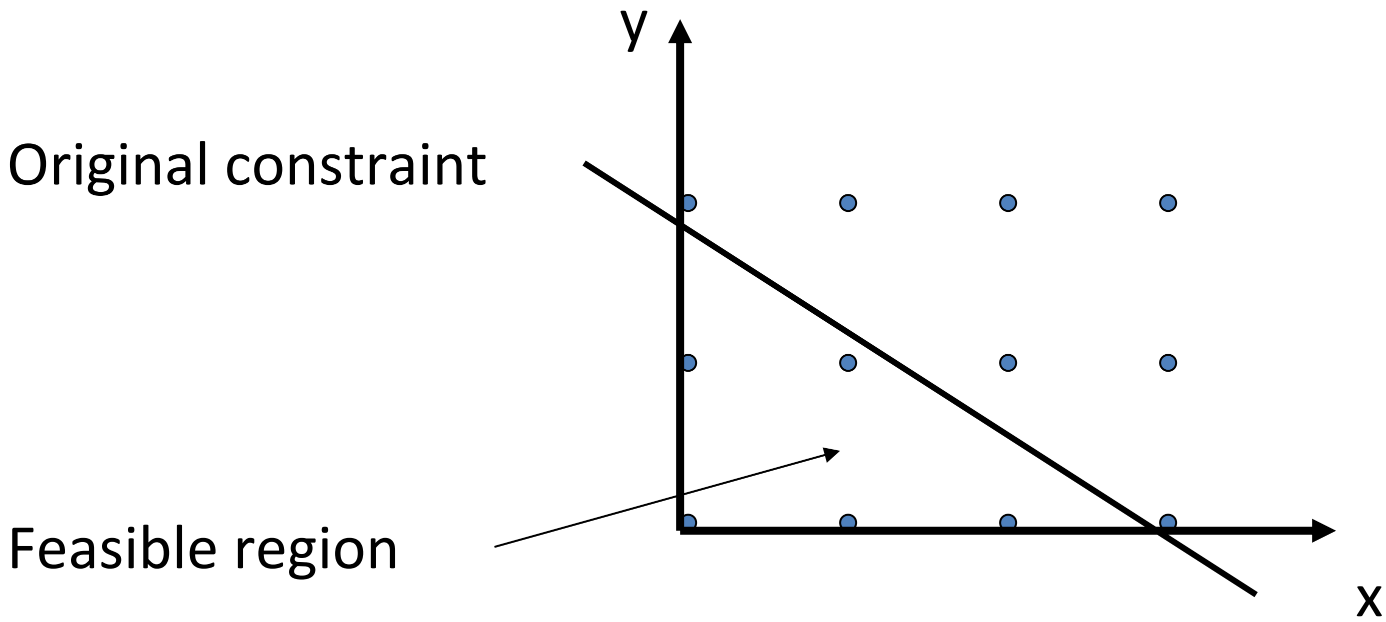
- Similar steps in both cases:
  - Add/replace constraints in model to tighten formulation
    - Same integer solutions
    - Fewer continuous solutions
- Important difference:
  - Presolve is applied to the original model to create a new model
  - Cutting planes are added to an existing model (typically the presolved model) to cut off a relaxation solution

Model Reformulation

Presolve versus cutting planes

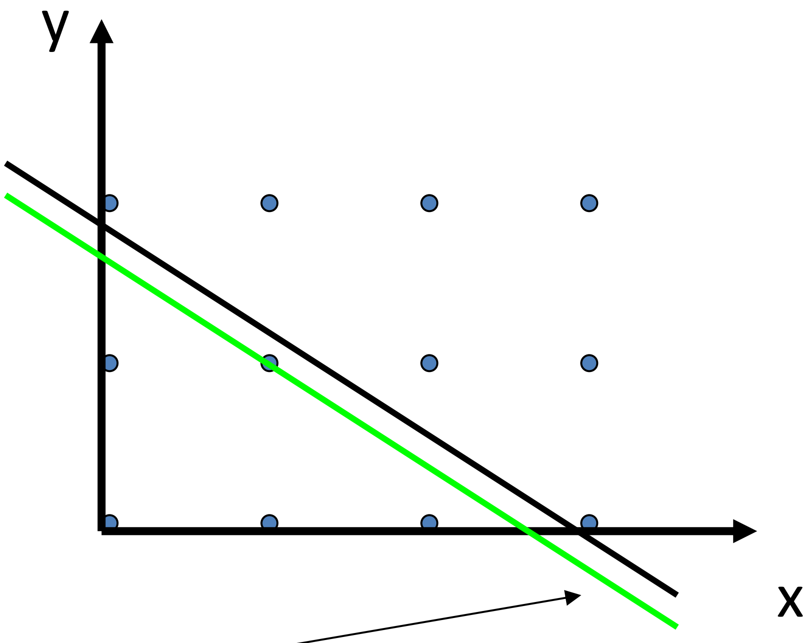
- Important difference:
  - A single constraint can produce an exponential number of tighter constraints
  - Presolve introduces tighter constraints that dominate existing constraints
    - Tighter formulation without creating a larger problem
    - Reformulation is independent of relaxation solution
  - Cutting planes introduce tighter constraints that cut off a particular relaxation solution
    - Focused growth in model size

Simple Reformulation Example



Simple Reformulation Example

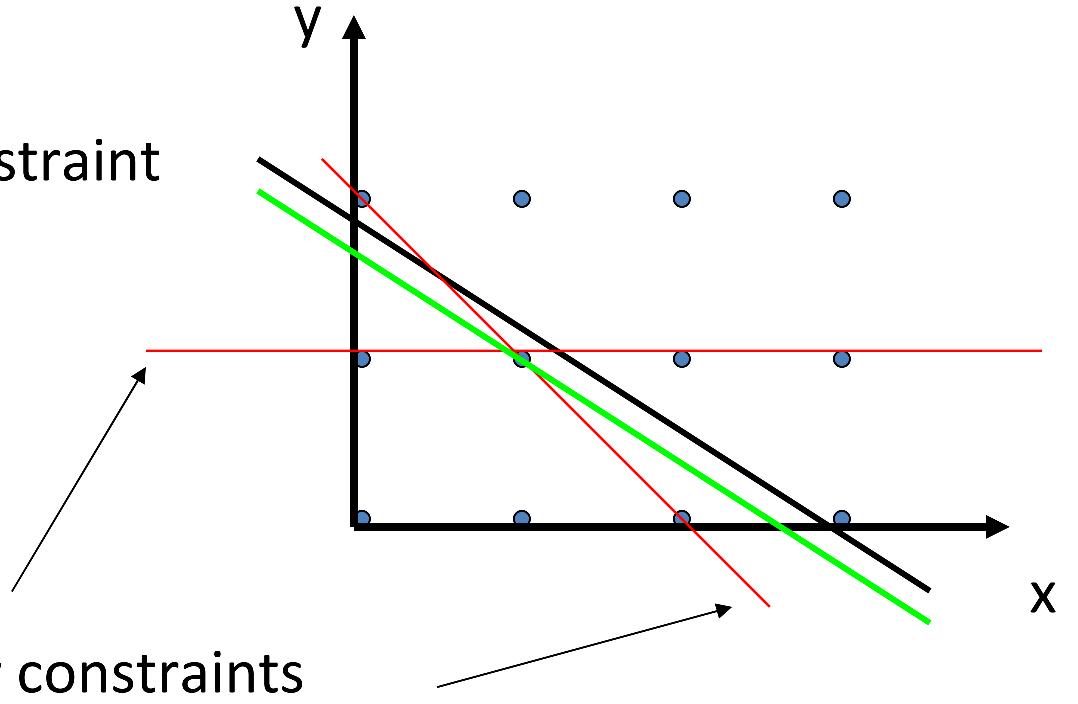
Original constraint



Tighter constraint

Simple Reformulation Example

Original constraint



Even tighter constraints

Rounding, Lifting, and Disjunction

- Three powerful, widely used concepts in presolve and cutting planes:
- Rounding
  - Integer multiples of integer variables take integer values
- Lifting
  - Fixing a binary variable at a bound may cause a constraint to go slack
- Disjunction
  - Binary variable must take one of two values

Rounding

Simplest Form of Rounding

Rounding in presolve

- A fractional bound on an integer variable can be truncated:
  - $x \leq 1.5$ implies $x \leq 1$
- Effects can become non-trivial when combined with bound strengthening:
  - $x + 2y + 4z = 4$, all variables binary
  - Bound strengthening and rounding together yield:
    - $4z \geq 4 - \sup(x+2y)$; $z \geq ¼$; z >= 1
    - x=0, y=0, z=1

GCD Reduction

More rounding in presolve

- Given a constraint involving all integer variables with integer coefficients
  - $\sum a_j x_j \leq b$
- Divide through by GCD of coefficients (g)
  - $\sum (a_j/g) x_j \leq \lfloor b/g \rfloor$
- LHS is integral, so RHS can be truncated
- Example:
  - $3x + 6y + 9z \leq 11$

Chvátal-Gomory Rounding Cut

Yet more rounding

- Given a constraint involving non-negative integer variables
  - $\sum a_j x_j \leq b$
- Divide through by some positive constant c
  - $\sum (a_j/c) x_j \leq b/c$
- Truncate coefficients
  - $\sum \lfloor a_j/c \rfloor x_j \leq \sum (a_j/c) x_j \leq b/c$
- LHS is integral, so RHS can be truncated
- Note: does not necessarily dominate original constraint
  - (Probably) not relevant for presolve

Gomory Rounding Cut

Example

- Given a constraint involving non-negative integer variables
  - $3x + 3y + 5z \leq 8$
- And relaxation solution:
  - $x=1, y=1, z=2/5$
- Divide through by 3
  - $x + y + 5/3\, z \leq 8/3$
- Truncate coefficients and RHS
  - $x + y + z \leq 2$
- Cuts off relaxation solution:
  - $x + y + z = 12/5$

Lifting

Coefficient Reduction

Lifting in presolve

- Given a constraint involving some binary x_k:
  - $\sum a_j x_j \geq b$
- Will fixing $x_k=1$ cause constraint to go slack?
  - $a_k + \inf (\sum_{j!=k} a_j x_j) > b$?
  - $s = a_k + \inf (\sum_{j!=k} a_j x_j) - b > 0$
- If so, we can subtract the following from LHS:
  - $s x_k$
- Example:
  - $2x + y \geq 1$ becomes
  - $x + y \geq 1$

Implied Bound Cuts

Trivial lifting for cutting planes

- Given a continuous variable with an upper bound
  - $y \leq u$
- And a binary variable x that implies a new upper bound on y:
  - e.g., x=0 -> $y \leq u_i$
- Can lift x into '$y \leq u$'
  - $y + (u-u_i)(1-x) \leq u$
- Simple case: $u_i=0$
  - Cut: $y \leq u\,x$

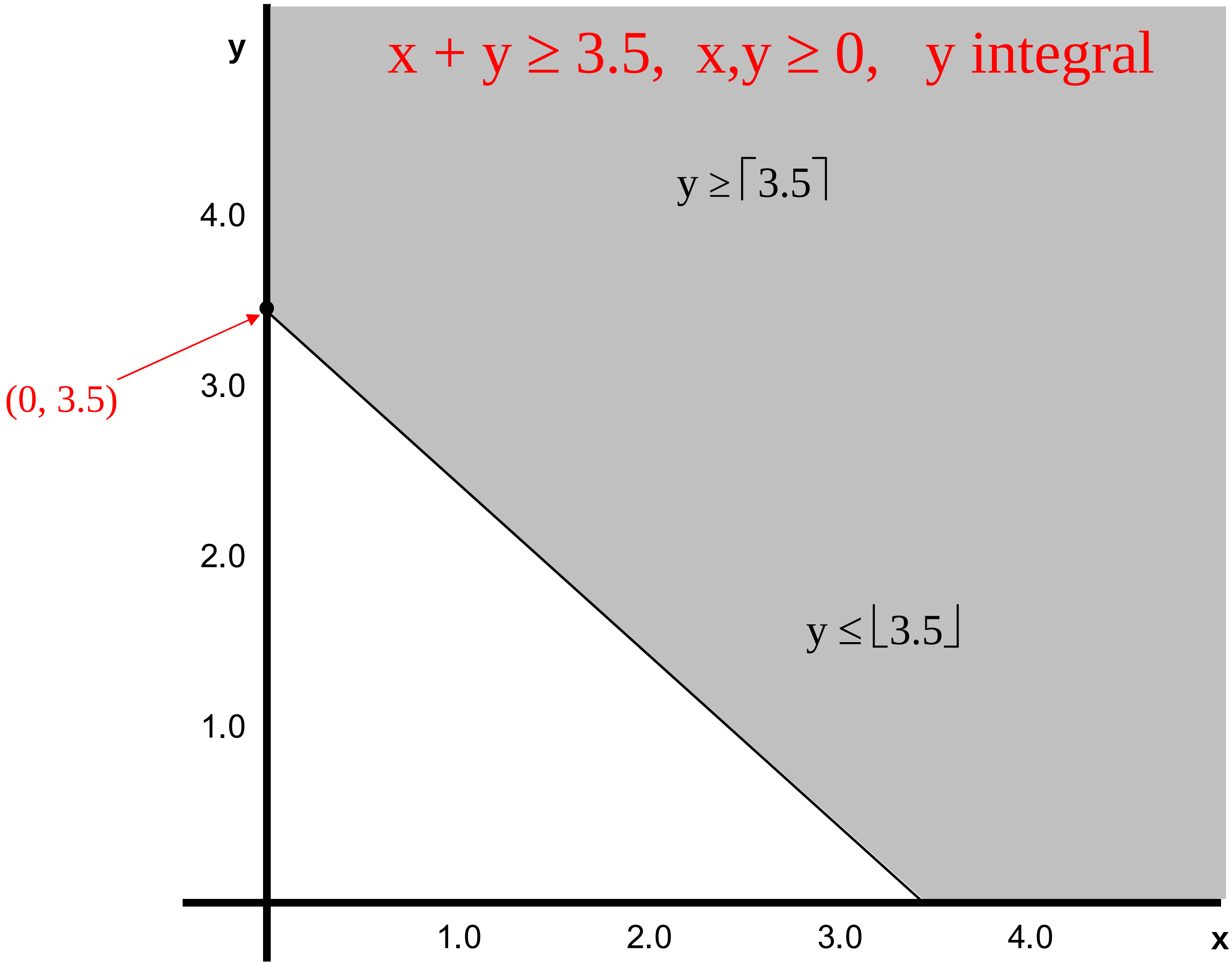
Implied Bound Cuts

Example

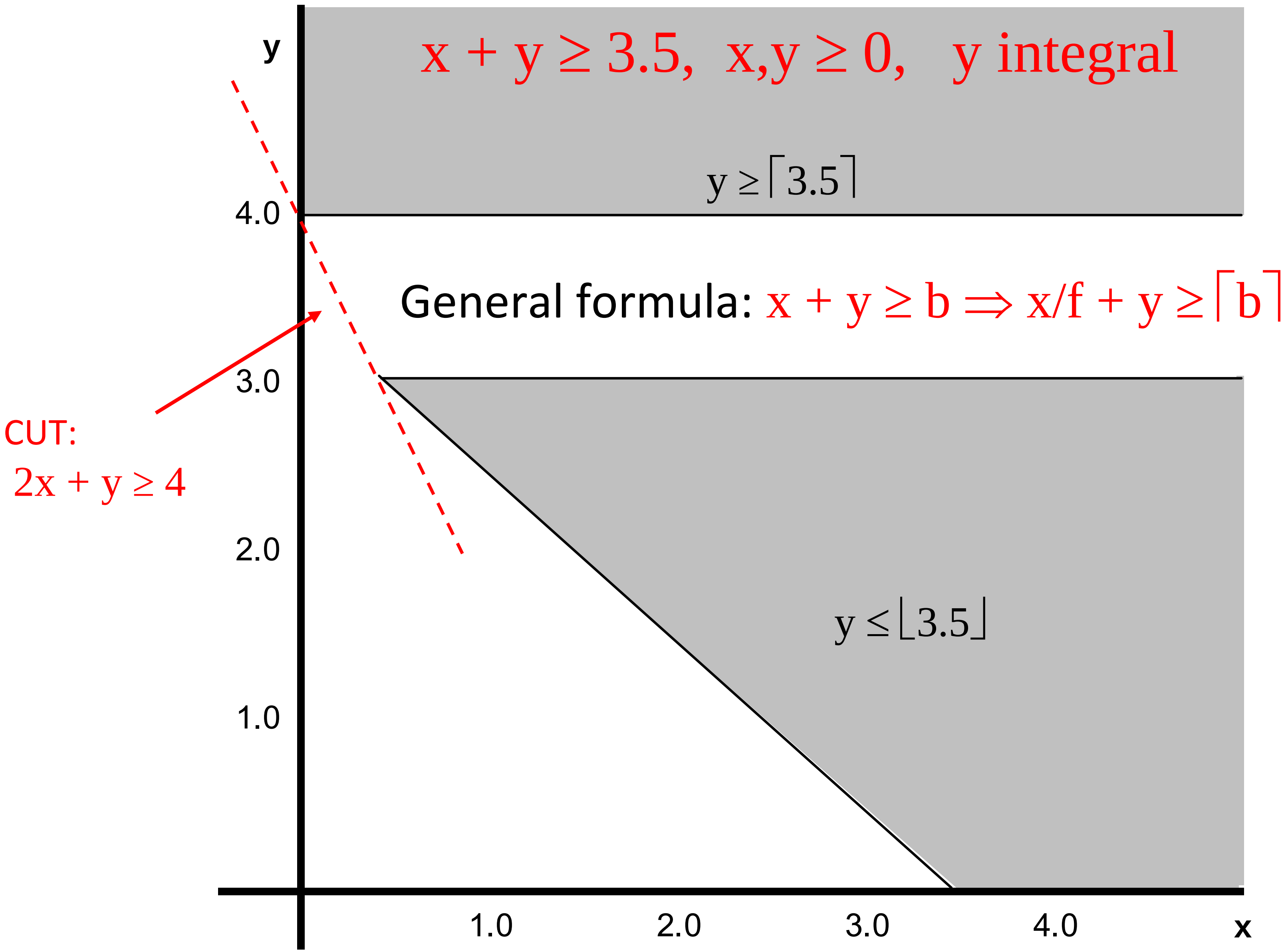
- Given continuous variables with upper bounds
  - $y_1 + y_2 \leq 10\, x$
  - $y_1 \leq 5$ and $y_2 \leq 5$
- Implied bound cuts:
  - $y_1 \leq 5\, x$
  - $y_2 \leq 5\, x$
- Violated by relaxation solution

Disjunction

Mixed-Integer Rounding Cut



Mixed-Integer Rounding Cut



Gomory Mixed Cut

- Given $y, x_j \in Z_+$, and

  $y + \sum a_{ij}x_j = d = \lfloor d \rfloor + f, \; f > 0$
- Rounding: Where $a_{ij} = \lfloor a_{ij} \rfloor + f_j$, define

  $t = y + \sum(\lfloor a_{ij} \rfloor x_j : f_j \leq f) + \sum(\lceil a_{ij} \rceil x_j : f_j > f) \in Z$
- Then

  $\sum(f_j \, x_j : f_j \leq f) + \sum(f_j - 1)x_j : f_j > f) = d - t$
- Disjunction:

  $t \leq \lfloor d \rfloor \Rightarrow \sum(f_j x_j : f_j \leq f) \geq f$

  $t \geq \lceil d \rceil \Rightarrow \sum((1-f_j)x_j : f_j > f) \geq 1-f$
- Combining:

  $\sum((f_j/f)x_j : f_j \leq f) + \sum([(1-f_j)/(1-f)]x_j : f_j > f) \geq 1$

Computing Gomory Mixed Cuts

1. Make a an ordered list of "sufficiently" fractional variables.
2. Take the first 100. Compute corresponding "tableau" rows. Reject if coefficient range too big.
3. Add to LP.
4. Repeat twice.
5. Computed only at root. Slack cuts purged at end of root computation.

Other Presolve Techniques

Problem Size Reductions

More Presolve Reductions

- Fixed variables
- Inactive constraints:
  - Example: $x + y \leq 2$, x and y binary
- Redundant constraints:
  - Example: $x + y \leq 2$; $x + y \leq 3$
- Dual fixed reductions:
  - Variable with:
    - Positive objective coefficient (minimization problem)
    - Belonging to only less-than-constraints
    - Having all non-negative matrix coefficients
  - …can be fixed to lower bound

Presolve Summary

- Presolve a vital part of solving a MIP model
- Most models have significant scope for improvement
  - 5X+ problem size reductions are common
  - 10X runtime reductions are typical

Other Cutting Plane Techniques

Cover (Knapsack) Cuts

- 0-1 Knapsack

  $K = \{x \in \{0,1\}^N: \sum_{j \in N} a_j x_j \leq b\}$, with $a_j > 0$ and $b > 0$

- The set $C \subseteq N$ is called a *cover* if

  $\sum_{j \in C} a_j > b$

- The cover inequality

  $\sum_{j \in C} x_j \leq |C| - 1$

  is valid for K

Cover+Lifting: 0-1 Knapsack

- Consider

  $$5x_1 + 5x_2 + 5x_3 + 5x_4 + 3x_5 + 8x_6 \leq 17$$

- Cover inequality (in fact, this cover is minimal)

  $$x_1 + x_2 + x_3 + x_4 \leq 3$$

- Lifting x_5 first, then x_6

  $$x_1 + x_2 + x_3 + x_4 + \pi_5 x_5 \leq 3$$

  $$\pi_5 = 3 - \max \{x_1 + x_2 + x_3 + x_4 : x_5 = 1\} = 1$$

  Similarly, $\pi_6 = 1$, so the lifted cover is

  $$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$$

- Lifting x_6 first, then $x_{5,}$ then the lifted cover is

  $$x_1 + x_2 + x_3 + x_4 + 2x_6 \leq 3$$

Clique Cuts

- Two binary variables are *incompatible* if they can't both be 1:
  - $x + y \leq 1$ means x and y are incompatible
- A *clique* is a set of pairwise incompatible variables
  - $C = \{x \in B: x_i \text{ and } x_j \text{ are incompatible}\}$
  - Clique cut: $\sum_{j \in C} x_j \leq 1$
- Example:
  - $x + y \leq 1$; $x + z \leq 1$; $y + z \leq 1$ implies
  - $x + y + z \leq 1$

Cutting Plane Summary

Sample Gurobi Output

```
Read MPS format model from file mzzv11.mps.bz2
Optimize a model with 9499 Rows, 10240 Columns and 134603 NonZeroes
Presolved: 8130 Rows, 8524 Columns, 124722 Nonzeros
Objective GCD is 2
Found heuristic solution: objective -1120.000

Root relaxation: objective -2.276189e+04, 9075 iterations, 3.97 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time
     0     0  -22761.89    0  470  -1120.000  -22761.89  1932%     -    6s
H    0     0                       -10530.00  -22761.89   116%     -    6s
H    0     0                       -21278.00  -21957.07  3.19%     -   10s
     0     0  -21947.20    0  469  -21278.00  -21947.20  3.15%     -   10s
     0     0  -21940.37    0  576  -21278.00  -21940.37  3.11%     -   10s
….
     0     2  -21938.57    0  422  -21278.00  -21938.57  3.10%     -   21s
H   28    20                       -21388.00  -21846.34  2.14%   102   25s
H   60    26                       -21718.00  -21846.34  0.59%  71.8   27s
   171    13     cutoff   10        -21718.00  -21801.53  0.38%  49.2   31s
   693    51  -21730.73   34   55  -21718.00  -21735.52  0.08%  29.3   36s

Cutting planes:
  Gomory: 10
  Cover: 23
  Implied bound: 81
  Clique: 11
  MIR: 13
  Flow cover: 40

Explored 873 nodes (49165 simplex iterations) in 37.91 seconds

Optimal solution found (tolerance 1.00e-04)
Best objective -2.1718000000e+04, best bound -2.1718000000e+04, gap 0.0000%
```

Applying Cutting Planes

- Many different varieties of cutting planes
- Number that are valid for a particular model is enormous
  - Must identify relevant ones
- Must solve the *separation* problem to find violated cutting planes
  - Heuristic procedure for each type of cutting plane (not discussed)

Applying Cutting Planes

- How many cuts should be generated for a relaxation solution?
  - One?
    - Will provide a new relaxation solution
    - Expensive to re-solve relaxation for each cut
  - As many as possible?
    - Relaxation solution only needs to be cut off once
    - Cuts increase the size of the model
- Need to strike a balance
  - Multiple rounds of cutting plane generation
  - Limited number of cuts per round

Cutting Planes in Gurobi V1.1

- Gomory
  - Gu ISMP 2006 , strengthen by lifting in GUBs
- Flow covers
  - Strengthened by lifting before un-transforming
- MIR
  - Different aggregation for MIR and flow covers
- Knapsack covers
- Clique
- Implied bound
- Flow paths
  - Results fed into flow covers
- GUB covers

Some Computational Results

Computational Results: 106 Models

Which Single Feature Helps Most?

(CPLEX 8.0 < 1000 seconds, 5.0 unsolvable)

Cuts	53.7x
Presolve	10.8x
CPLEX 5.0 presolve	3.1x
CPLEX 5.0 var. selection	2.9x
No heuristics	1.4x
No node presolve	1.3x

Computational Results IV: 106 Models

Removing Single Cuts

• Gomory mixed-integer	2.52x
• Mixed-integer rounding	1.83x
• Knapsack cover	1.40x
• Flow cover	1.22x
• Implied bound	1.19x
• Path	1.04x
• Clique	1.02x
• GUB cover	1.02x
• Disjunctive	0.53x

Parallel MIP

Why Parallel?

- Microprocessor trends have changed
- Transistors are:
  - Still getting smaller
  - But not faster
- Implications:
  - New math for CPUs: more transistors = more cores
    - 4 cores now, 8 cores in late 2009, …
  - *Sequential software won't be getting significantly faster in the foreseeable future*
- Gurobi MIP solver built for parallel from the ground up
  - Sequential is just a special case

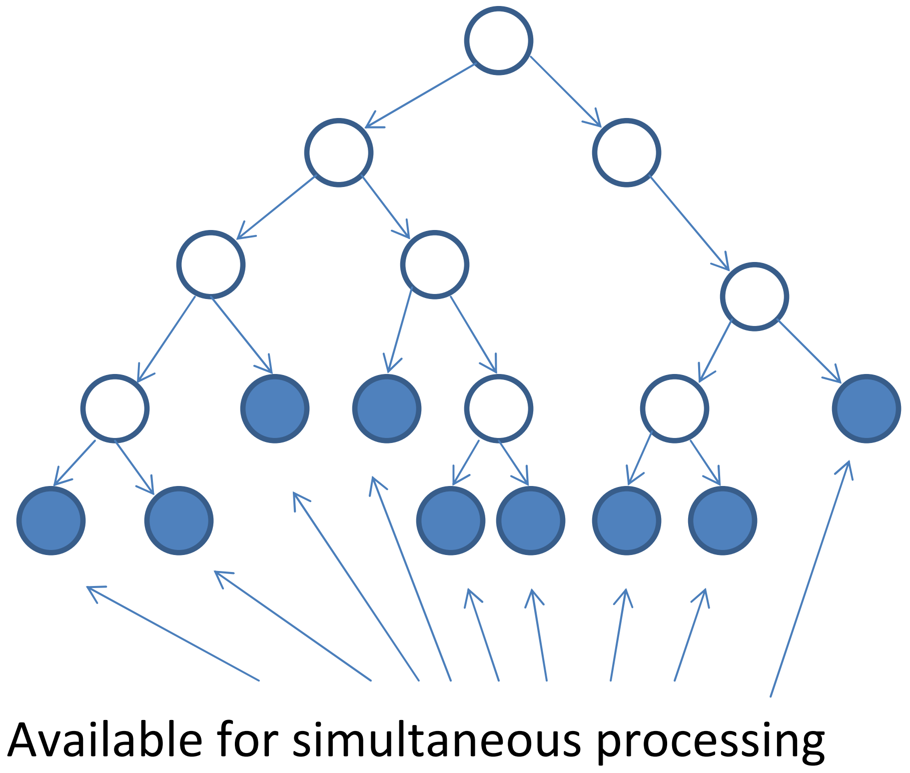
Need Deterministic Behavior

- Non-deterministic parallel behavior:
  - Multiple runs with the same inputs can give different results
- "Insanity: doing the same thing over and over again and expecting different results."
  - Albert Einstein
- Conclusion: non-deterministic parallel behavior will drive you insane

Building Blocks

Building Blocks

- Parallel MIP is parallel branch-and-bound:



Deterministic Parallel MIP

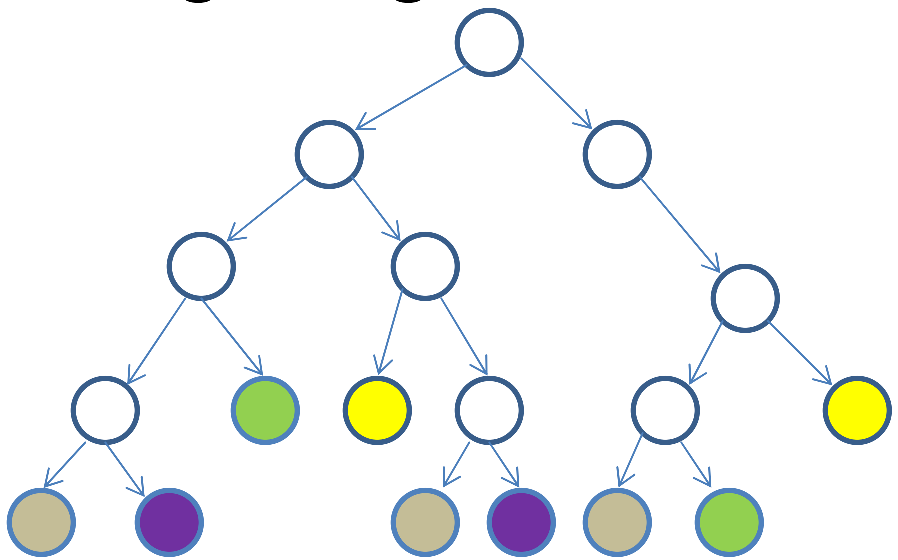
- Multiple phases
- In each phase, on each processor:
  - Explore nodes assigned to processor
  - Report back results
    - New active nodes
    - New solutions
    - New cuts
    - Etc.
  - Must have a clear distinction between local and global information
- One approach to node assignment:
  - Assign a subtree to each processor
  - Limit amount of exploration in each phase

Subtree Partitioning

- Problem:
  - Subtree may quickly prove to be uninteresting
    - Poor relaxation objectives
      - May want to abandon it
    - Pruned quickly
      - Leaves processor idle

More Global Partitioning

- Node coloring: assign a color to every node



  - Processor can only process nodes of the appropriate color
  - New child node same color as parent node
  - Perform periodic re-coloring

More Dynamic Node Processing

- Allows much more flexibility
  - Processor can choose from among many nodes of the appropriate color
- *Deterministic priority queue* data structure required to support node coloring
  - Single global view of active nodes
  - Support notion of node color
    - Processor only receives node of the appropriate color
  - Efficient, frequent node reallocation
  - Achieves much better load balancing

Performance

Performance Benchmarks Gurobi V1.1

- Performance test sets:
  - Mittelmann optimality test set:
    - 55 models, varying degrees of difficulty
    - http://plato.asu.edu/ftp/milpc.html
  - Mittelmann feasibility test set:
    - 34 models, difficult to find feasible solutions
    - http://plato.asu.edu/ftp/feas_bench.html
  - Our own broader test set:
    - A set of 263 models that require between one minute and one hour to solve on one core
      - Publicly available models, plus a few customer models
- Test platform:
  - Q9450 (2.66 GHz, quad-core system)

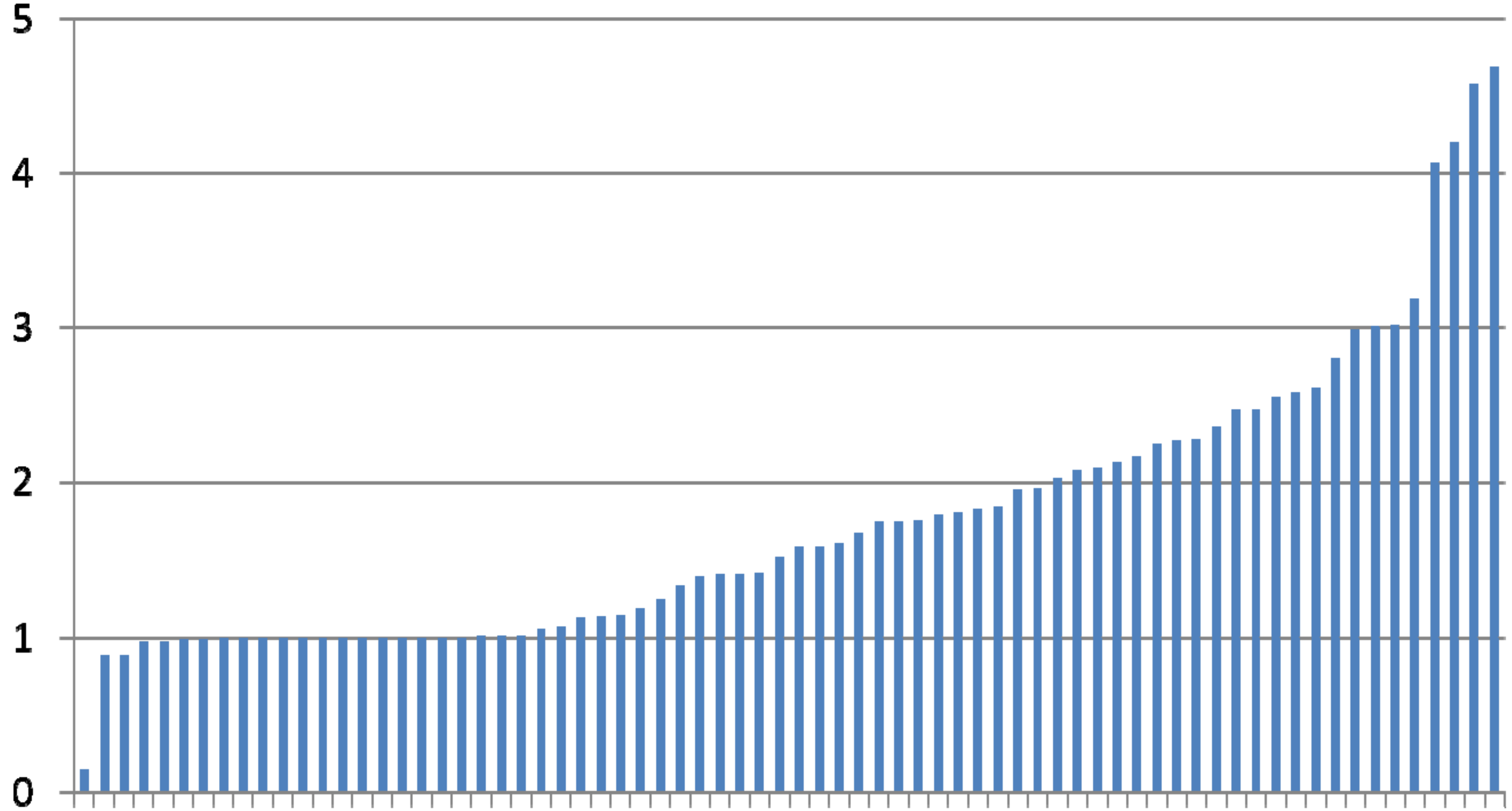
Performance Results

- Two basic questions:
  - Is P=1 efficient?
  - How much does performance improve with P>1?
- P=1 efficiency (geometric means):
  - Mittelmann optimality test set
    - Gurobi 1.1 is 1.18X **slower** than CPLEX 12.0
  - Mittelmann feasibility test set
    - Gurobi 1.1 is 2.3X **faster** than CPLEX 12.0

Parallel Performance

- Comparisons:
  - Gurobi
    - 1.45X **faster** than CPLEX 12.0 for p=4 on Mittelmann
    - Gurobi p=1 versus p=4
      - 1.73X speedup on Mittelmann
      - 1.61X speedup on broader set
  - CPLEX p=1 versus p=4
    - 1.21X speedup for CPLEX 12 on Mittelmann (non-deterministic)
    - 1.29X speedup for CPLEX 11 on their broader set

Speedups for P=4 (By Model)



Gurobi 2.0

- Improve simplex solver
- Cutting planes
  - Add missing cutting planes
- Add missing presolve reductions