GUROBI OPTIMIZATION

Gurobi 最优算法和启发式算法的融合

顾宗浩

GUROBI CTO, 联合创始人

深圳, 2018年7月13日

Outline

- MIP Algorithm
- Heuristics and (~~vs.~~) optimization
- Gurobi heuristics
  - Non-LP based
  - LP based
  - Reformulation
  - Improvement
  - SubMIP and recursive
  - Features helping heuristics
- Gurobi heuristic parameters
- User input for Gurobi heuristics
  - MIP start/Multiple MIP starts
  - MIP hint
  - Partition heuristic
  - Heuristic callback
- What to do with too big/hard models

概要

- MIP算法
- 启发式和（~~vs.~~）优化
- Gurobi启发式算法
  - 基于非LP
  - 基于LP
  - 模型改建
  - 改进型
  - 子MIP和递归
  - 有助于启发式的功能
- Gurobi 启发式参数
- Gurobi启发式的用户输入功能
  - MIP 起始值/多个 MIP起始值
  - MIP提示
  - 分区启发式
  - 启发式回调
- 如何处理太大/太难的模型

Gurobi MIP Algorithms

Gurobi 混合整数规划算法

MIP Building Blocks 模块

- Presolve ⇦ Presolve, PrePasses, AggFill, Aggregate, DualReductions, PreSparsify, ... 预优化
  - Tighten formulation and reduce problem size
- Solve continuous relaxations ⇦ Method, NodeMethod 求解连续松弛模型
  - Ignoring integrality
  - Gives a bound on the optimal integral objective
- Cutting planes ⇦ Cuts, CutPasses, CutAggPasses, GomoryPasses, CliqueCuts, CoverCuts, FlowCoverCuts, ... 切平面
  - Cut off relaxation solutions
- Branching variable selection ⇦ VarBranch 分支变量
  - Crucial for limiting search tree size
- Primal heuristics ⇦ Heuristics, MinRelNodes, PumpPasses, RINS, SubMIPNodes, ZeroObjNodes 启发算法
  - Find integer feasible solutions

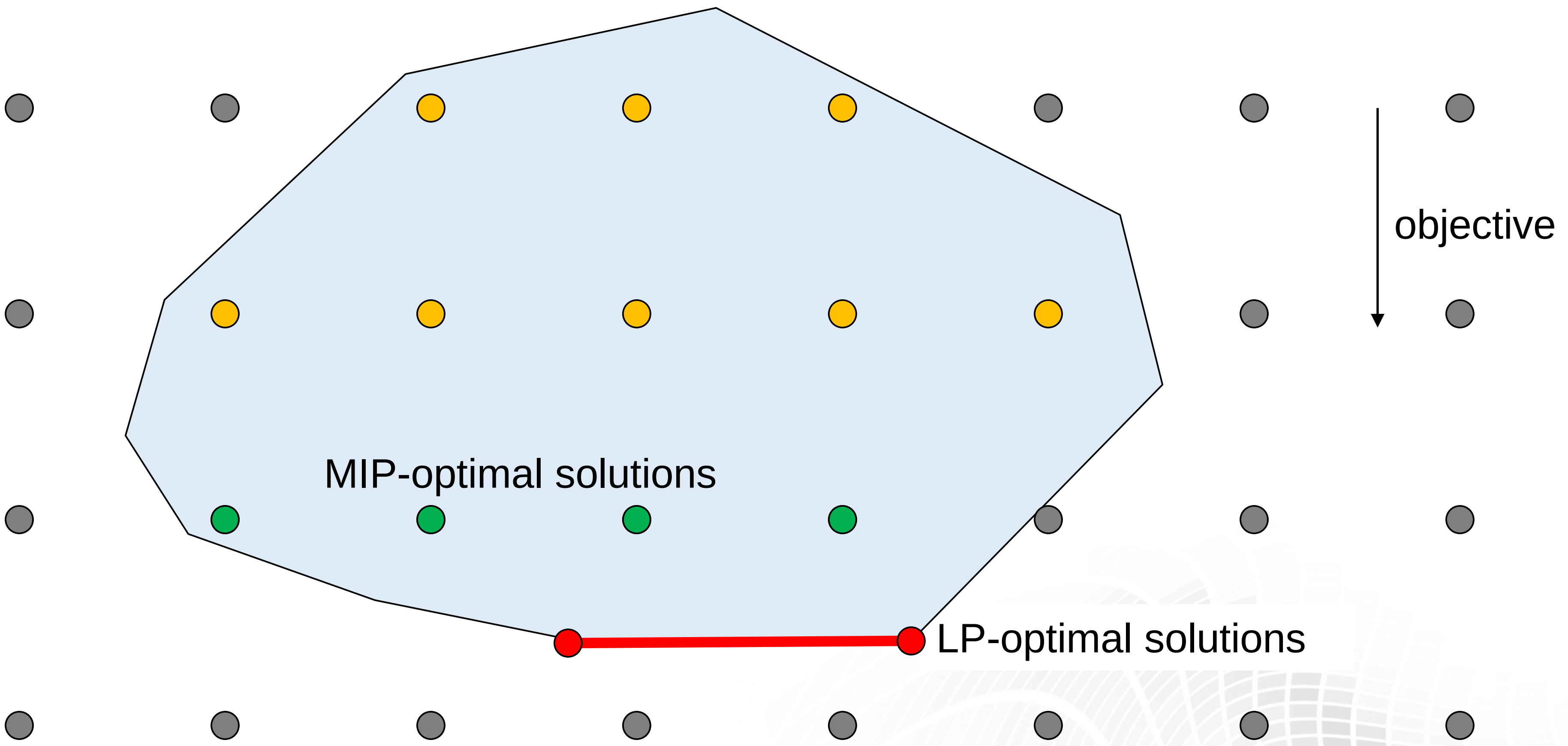
MIP Presolve 预优化

- Goals: 目标
  - Reduce problem size
  - Strengthen LP relaxation
  - Identify problem sub-structures
    - Cliques, implied bounds, networks, disconnected components, ...

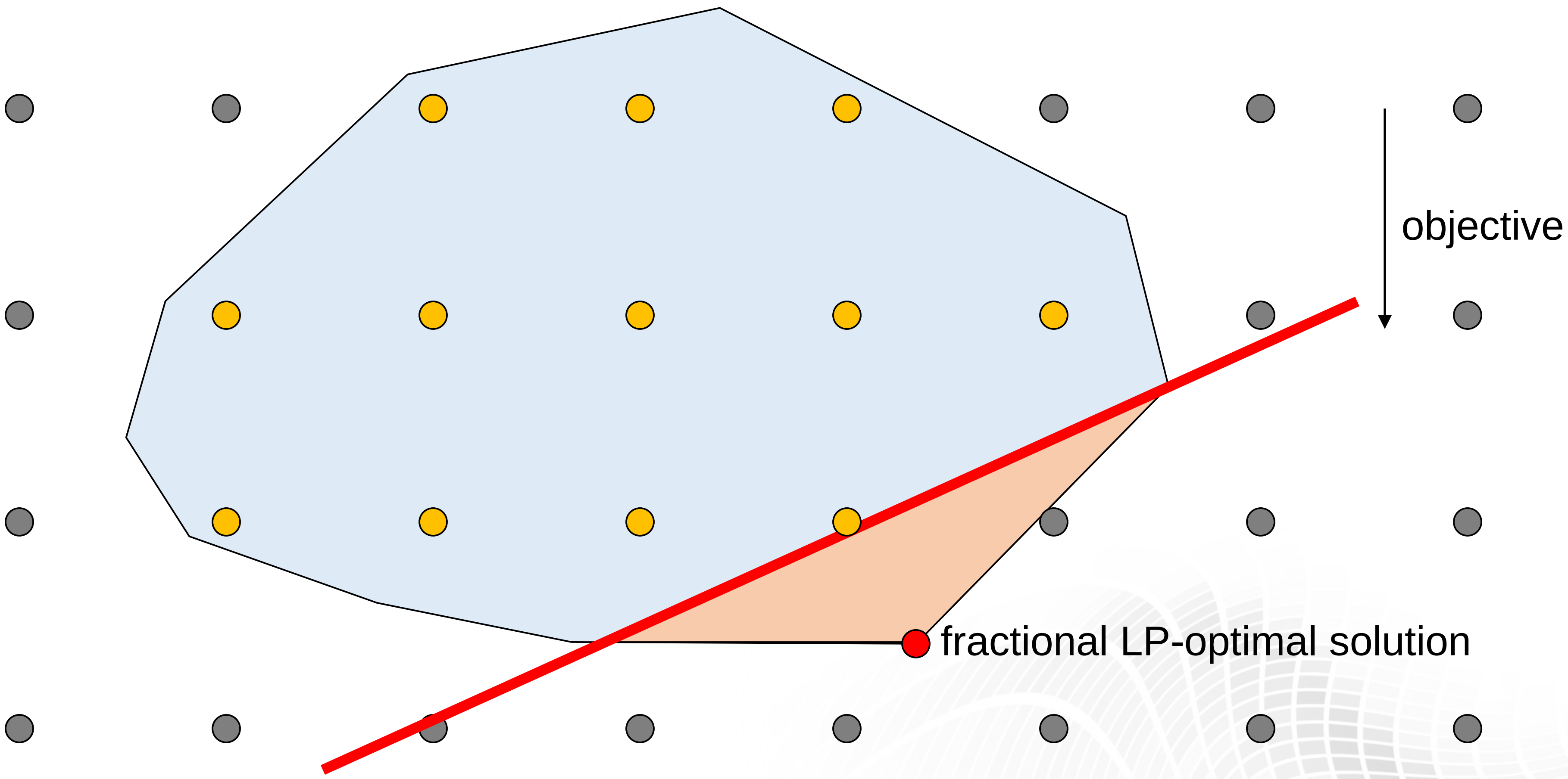
Disconnected

- Similar to LP presolve, but more powerful: 与LP 预优化类似，但更强大
  - Exploit integrality
    - Round fractional bounds and right hand sides
    - Lifting/coefficient strengthening
    - Probing
  - Does not need to preserve duality
    - We only need to be able to uncrush a primal solution
    - Neither a dual solution nor a basis needs to be uncrushed

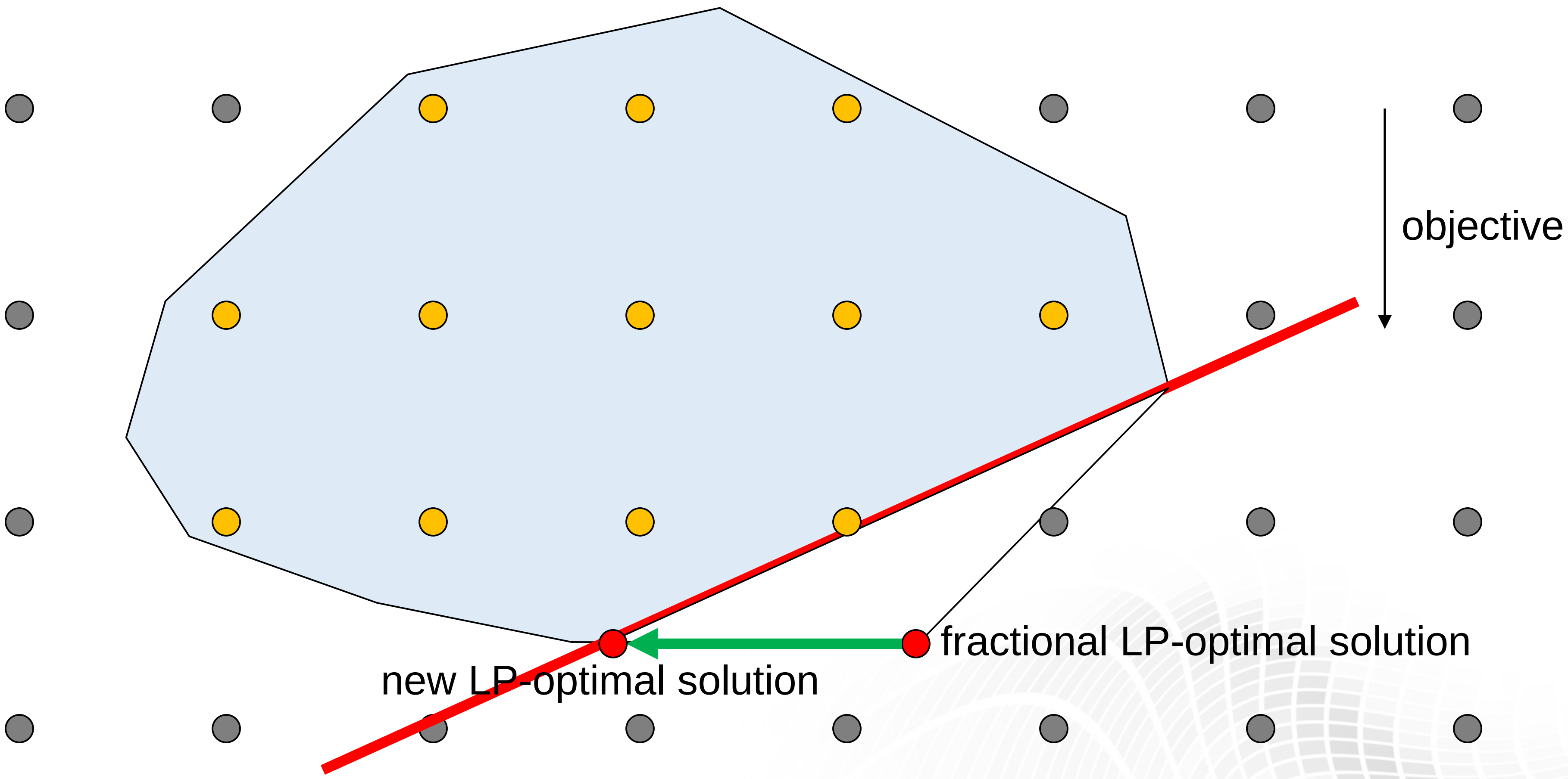
MIP – LP Relaxation　LP 松弛问题



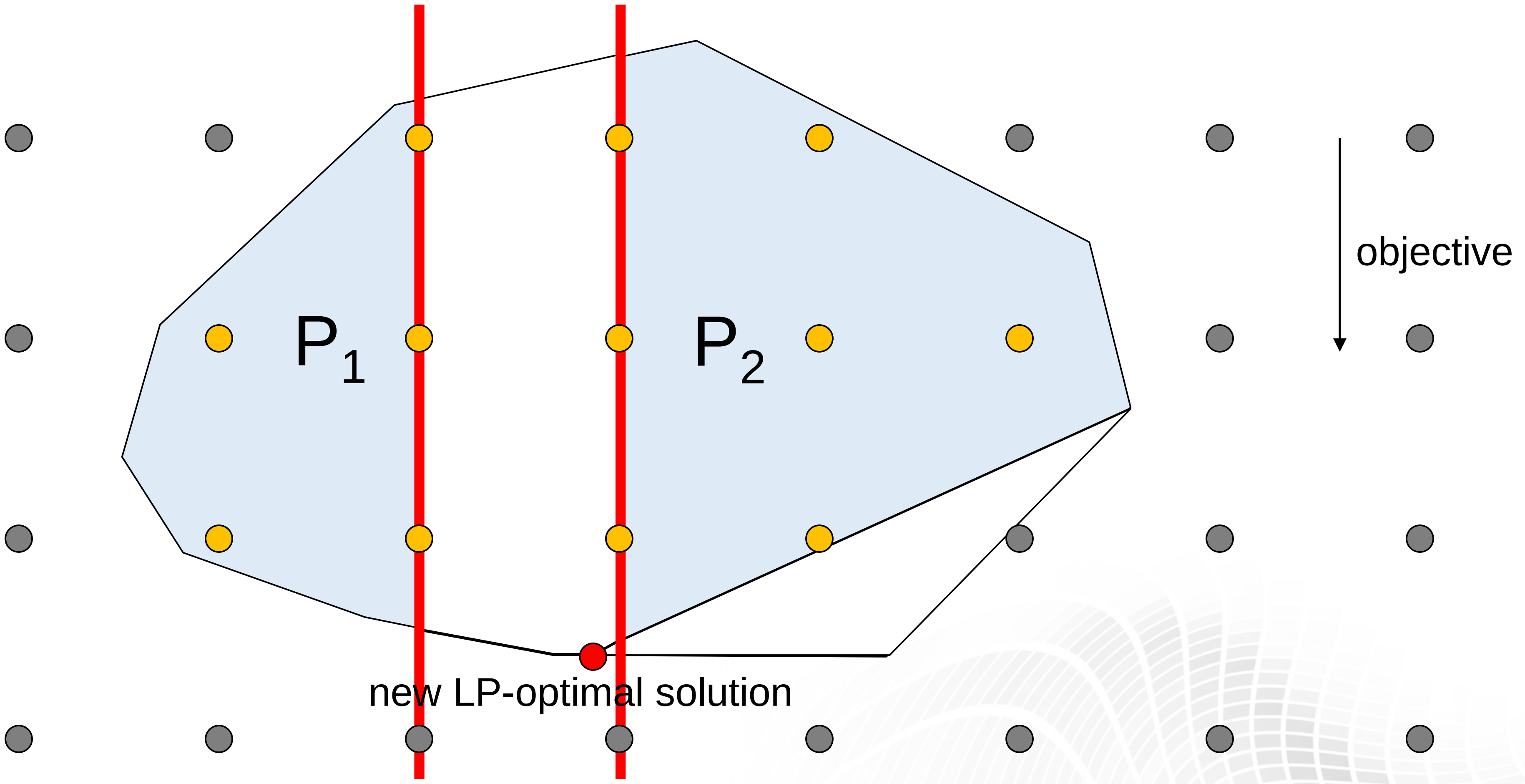
MIP – Cutting Planes 切平面



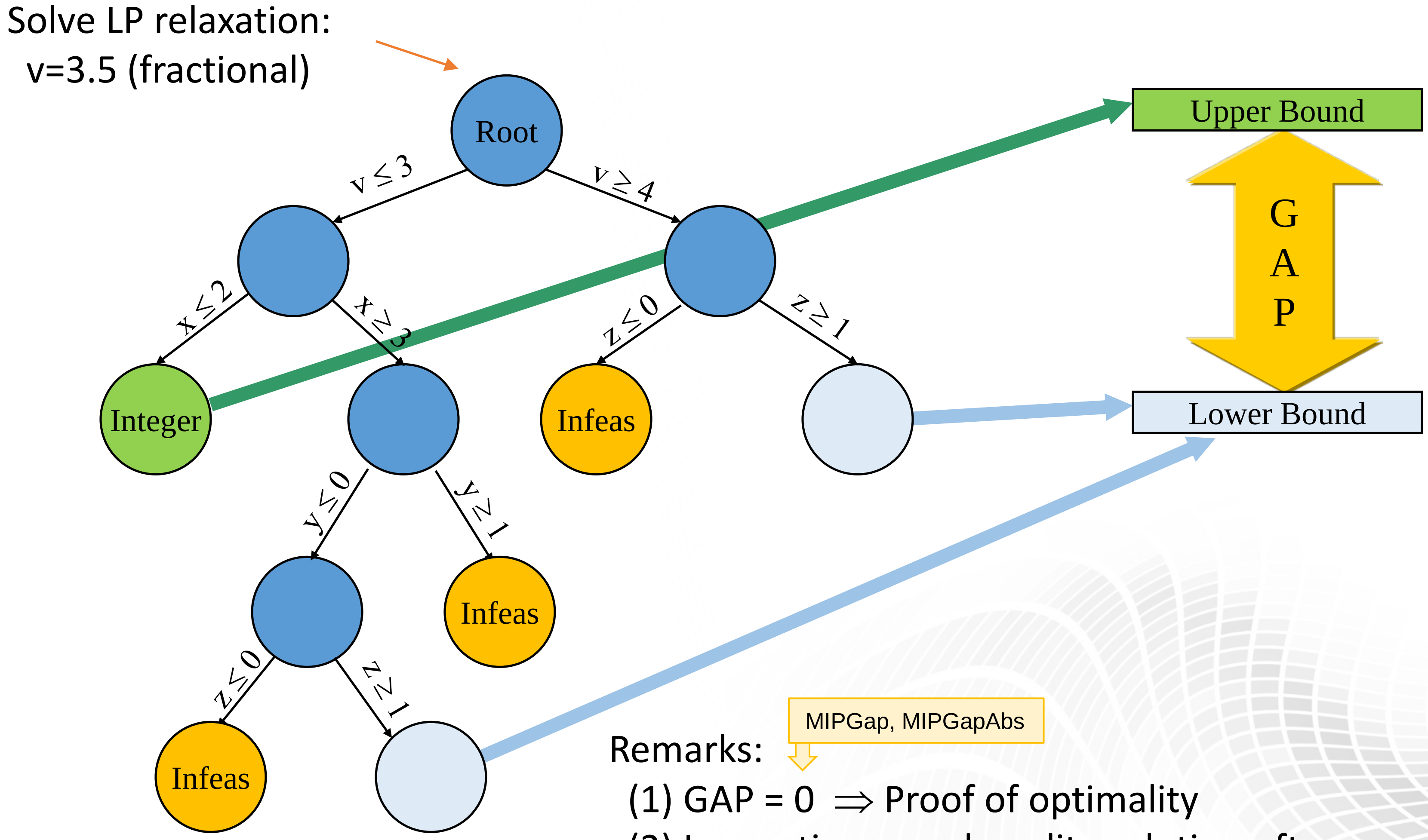
MIP – Cutting Planes 切平面



MIP – Branching 分支



LP based Branch-and-Bound 基于LP的分支定界法



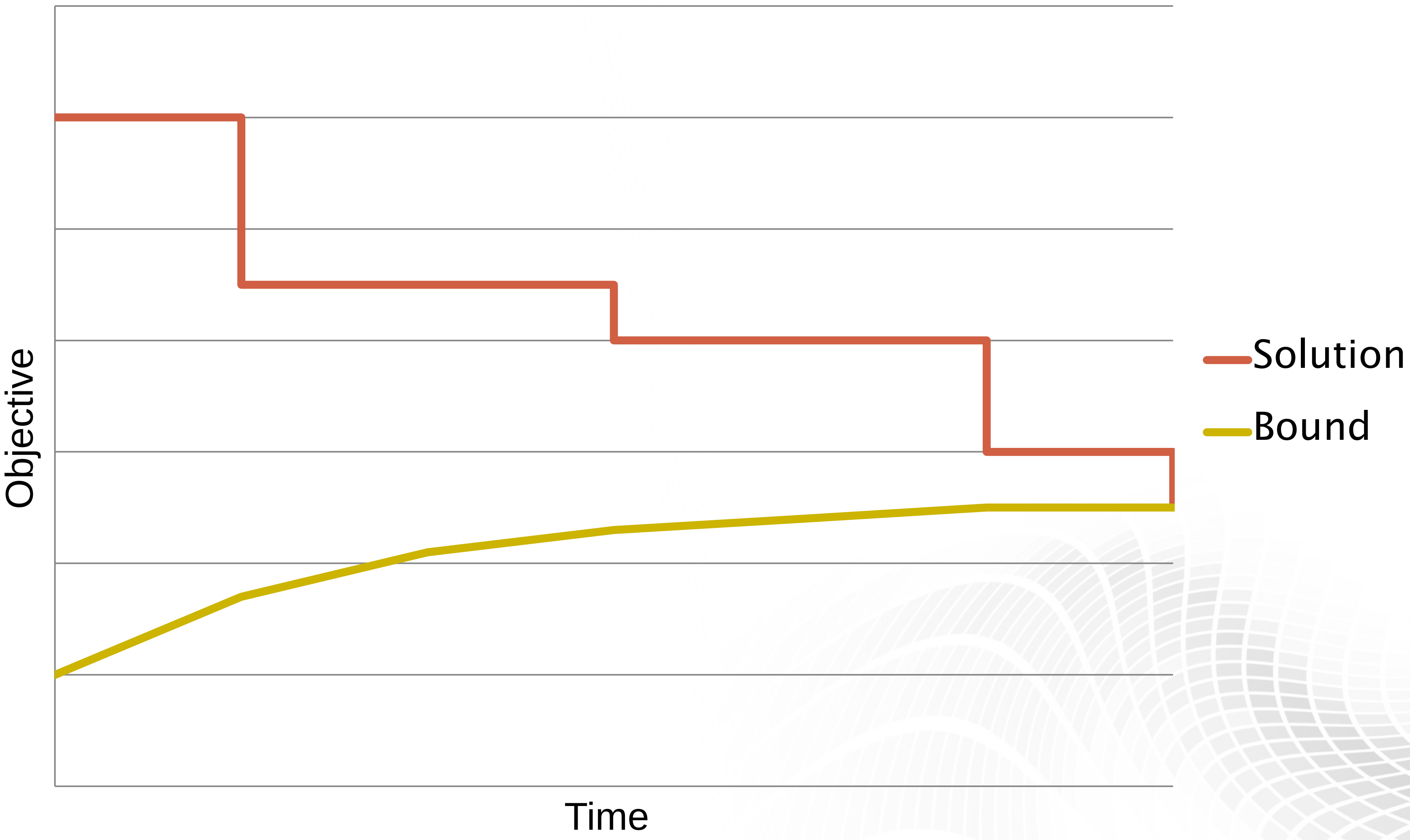
MIPGap, MIPGapAbs

Remarks:

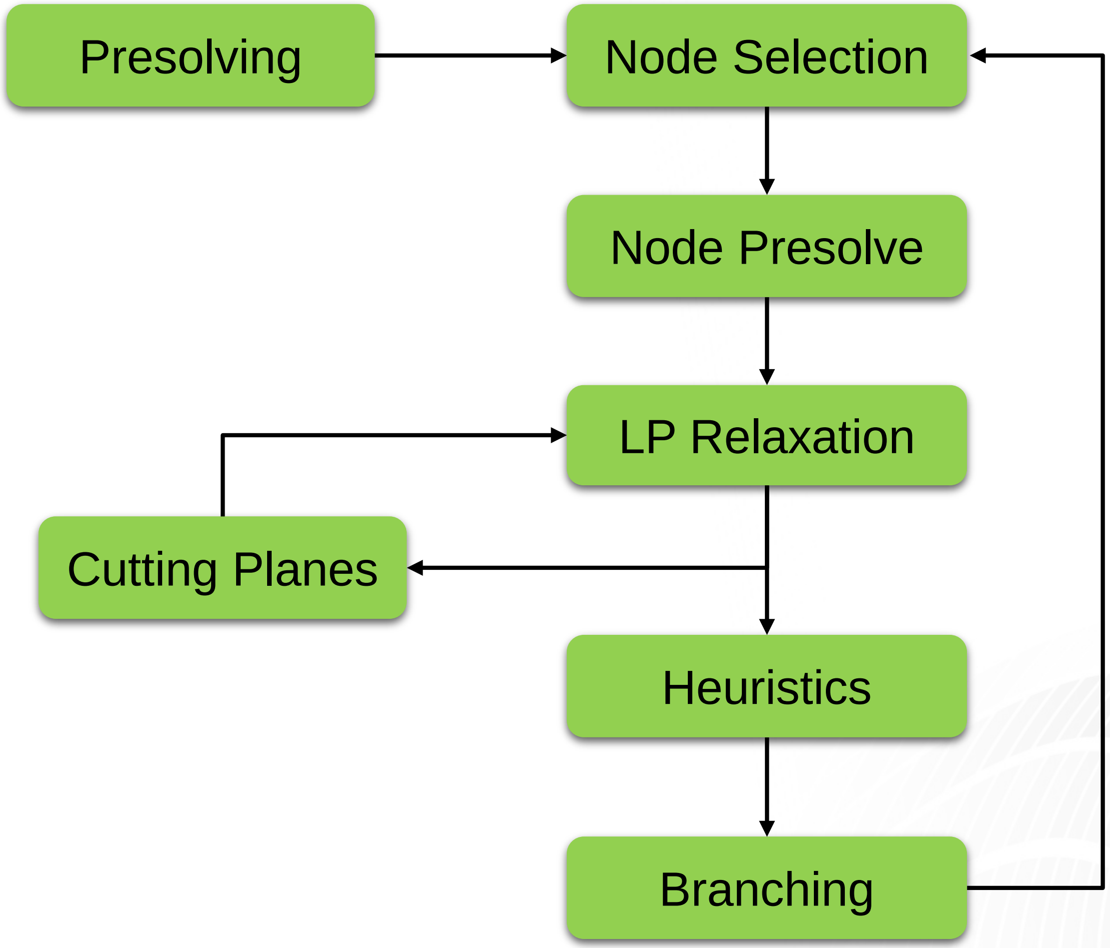
(1) GAP = 0 $\Rightarrow$ Proof of optimality

(2) In practice: good quality solution often enough

Solving a MIP Model 求解混合整数模型



Branch-and-Cut 分支切割



Branch-and-Cut 分支切割

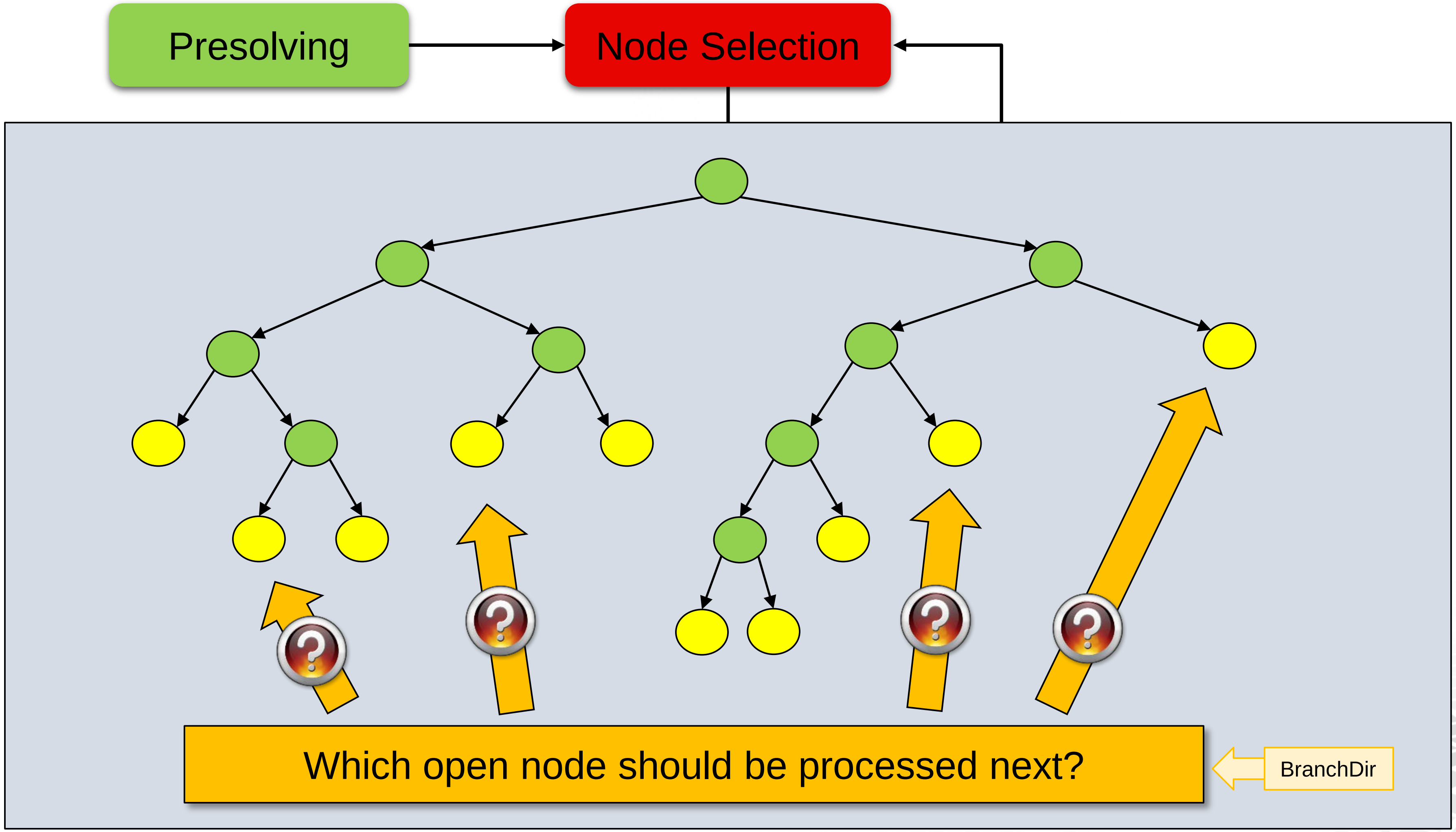
Presolving

Cutting Planes

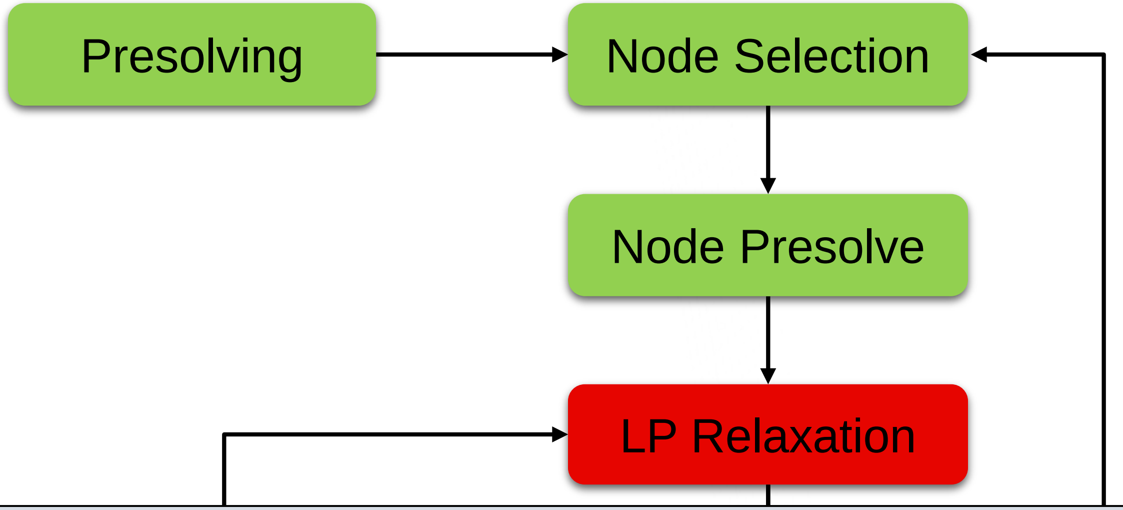
Branching

```
Gurobi Optimizer version 6.0.0 (linux64)
Copyright (c) 2014, Gurobi Optimization, Inc.

Read MPS format model from file /models/mip/roll3000.mps.bz2
Reading time = 0.03 seconds
roll3000: 2295 rows, 1166 columns, 29386 nonzeros
Optimize a model with 2295 rows, 1166 columns and 29386 nonzeros
Coefficient statistics:
  Matrix range    [2e-01, 3e+02]
  Objective range [1e+00, 1e+00]
  Bounds range    [1e+00, 1e+09]
  RHS range       [6e-01, 1e+03]
Presolve removed 1308 rows and 311 columns
Presolve time: 0.08s
Presolved: 987 rows, 855 columns, 19346 nonzeros
Variable types: 211 continuous, 644 integer (545 binary)

Root relaxation: objective 1.112003e+04, 1063 iterations, 0.03 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 11120.0279    0  154          - 11120.0279     -      -    0s
     0     0 11526.8918    0  207          - 11526.8918     -      -    0s
     0     0 11896.9710    0  190          - 11896.9710     -      -    0s
```

Branch-and-Cut 分支切割



Branch-and-Cut 分支切割




```
Presolved: 987 rows, 855 columns, 19346 nonzeros
Variable types: 211 continuous, 644 integer (545 binary)

Root relaxation: objective 1.112003e+04, 1063 iterations, 0.03 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 11120.0279    0  154          - 11120.0279     -      -    0s
     0     0 11526.8918    0  207          - 11526.8918     -      -    0s
     0     0 11896.9710    0  190          - 11896.9710     -      -    0s
...
H  327   218                    13135.000000 12455.2162  5.18%  42.6    1s
H  380   264                    13093.000000 12455.2162  4.87%  41.6    1s
H  413   286                    13087.000000 12455.2162  4.83%  41.4    1s
  1066   702 12956.2676   31  192 13087.0000 12629.5426  3.50%  [illegible]    5s
```

Branch-and-Cut 分支切割

Presolving

Cutting Planes

```
Presolved: 987 rows, 855 columns, 19346 nonzeros
Variable types: 211 continuous, 644 integer (545 binary)

Root relaxation: objective 1.112003e+04, 1063 iterations, 0.03 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 11120.0279    0  154          - 11120.0279     -      -    0s
     0     0 11526.8918    0  207          - 11526.8918     -      -    0s
     0     0 11896.9710    0  190          - 11896.9710     -      -    0s
     0     0 12151.4022    0  190          - 12151.4022     -      -    0s
     0     0 12278.3391    0  208          - 12278.3391     -      -    0s
...
  5485   634 12885.3652   52  143 12890.0000 12829.0134  0.47%  54.5   25s

Cutting planes:
  Learned: 4
  Gomory: 46
  Cover: 39
  Implied bound: 8
  Clique: 2
  MIR: 112
  Flow cover: 27
  GUB cover: 11
  Zero half: 91

Explored 6808 nodes (357915 simplex iterations) in 27.17 seconds
Thread count was 4 (of 8 available processors)
```

Branch-and-Cut分

```
Presolved: 987 rows, 855 columns, 19346 nonzeros
Variable types: 211 continuous, 644 integer (545 binary)

Root relaxation: objective 1.112003e+04, 1063 iterations, 0.03 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 11120.0279    0  154          - 11120.0279      -     -    0s
     0     0 11526.8918    0  207          - 11526.8918      -     -    0s
     0     0 11896.9710    0  190          - 11896.9710      -     -    0s
...
     0     0 12448.7684    0  181          - 12448.7684      -     -    0s
H    0     0                    16129.000000 12448.7684  22.8%     -    0s
H    0     0                    15890.000000 12448.7684  21.7%     -    0s
     0     2 12448.7684    0  181 15890.0000 12448.7684  21.7%     -    0s
H  142   129                    15738.000000 12450.7195  20.9%  43.8    1s
H  212   189                    14596.000000 12453.8870  14.7%  42.3    1s
H  217   181                    13354.000000 12453.8870  6.74%  42.6    1s
*  234   181              40    13319.000000 12453.8870  6.50%  42.1    1s
H  254   190                    13307.000000 12453.8870  6.41%  41.3    1s
H  284   194                    13183.000000 12453.8870  5.53%  42.6    1s
H  286   194                    13169.000000 12453.8870  5.43%  42.7    1s
```

Presolving

Cutting Pla

Heuristics

Branching

Branch-and-Cut分支切割

Presolvi

Cutting P

```
Presolved: 987 rows, 855 columns, 19346 nonzeros
Variable types: 211 continuous, 644 integer (545 binary)

Root relaxation: objective 1.112003e+04, 1063 iterations, 0.03 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 11120.0279    0  154          - 11120.0279     -      -    0s
     0     0 11526.8918    0  207          - 11526.8918     -      -    0s
     0     0 11896.9710    0  190          - 11896.9710     -      -    0s
...
H    0     0                    15890.000000 12448.7684  21.7%     -    0s
     0     2 12448.7684    0  181 15890.0000 12448.7684  21.7%     -    0s
...
  1066   702 12956.2676   31  192 13087.0000 12629.5426  3.50%  37.2    5s
  1097   724 12671.8285    8  147 13087.0000 12671.8285  3.17%  41.6   10s
  1135   710 12732.5601   32  126 12890.0000 12727.1362  1.26%  44.6   15s
  3416   887 12839.9880   46  136 12890.0000 12780.7059  0.85%  49.7   20s
  5485   634 12885.3652   52  143 12890.0000 12829.0134  0.47%  54.5   25s
```

Branching

Heuristics and (~~vs.~~) optimization 启发式和(~~对立~~)优化

- 优化是NP-hard
  - 我们只应考虑启发式，但可行性问题也是NP-hard
  - 理论上, 启发式和优化难度是一样的
- 许多实际问题经常被解到最优
- 我的优化问题很难解
  - 所以我只开发了自己的启发式算法
  - 那么你就是这次演讲的最佳听众
- 启发式算法和优化算法并不对立, 可以融合
- 我们将展示
  - Gurobi启发式的想法(ideas), 或许可以帮助您开发和改进您的启发式算法
  - 如何融合Gurobi优化算法和您的启发式算法为您找到更好的解决方案

GUROBI Heuristics

GUROBI启发式算法

Gurobi Heuristics 启发式算法

- Gurobi has more than 30 heuristics — 有30多种启发式算法
- Different types — 不同种类
  - Non-LP based — 基于非LP
    - Enumerate, search, greedy, …
  - LP based — 基于LP
    - Rounding, fixing & diving, …
  - Reformulation — 模型改建
    - Zero-objective, min relaxation, …
  - Improvement — 改进型
    - RINS ..
  - SubMIP and recursive — 子MIP和递归
    - Target heuristic, RINS, …
  - Problem specific — 针对具体问题的启发式
    - Fixed charge network heuristic
  - Features helping heuristics — 有助于启发式的功能
    - Pump reduce

Non-LP Based Heuristics: Greedy
基于非LP启发式算法: 贪婪算法

- Famous algorithms 着名算法
  - Optimal 最优的: shortest path最短的路径, min spanning tree 最小生成树
  - Not optimal非最优的: 0-1 Knapsack 背包

    Max 10 u + 8 v + 11 x + 7 y + 5 z

    s.t. 3 u + 4 v + 6 x + 4 y + 3 z ≤ 14

    u, v, x, y, z are binary variables

    Sorting variables based on the ratio of the obj. coefficient and constraint coefficient, already sorted

    Setting variables to one based on the order until it become infeasible. Here setting y to one become infeasible

    So the solution is (u, v, x, y, z) = (1, 1, 1, 0, 0) with obj. value 29

    Optimal solution (u, v, x, y, z) = (1, 1, 0, 1, 1) with obj. value 30
- Gurobi blind heuristics 盲目启发式 (blind means not using LP relaxation solution)
  - Sort binary/integer variables based on some measure 用某种度量对二进制/整数变量进行排序
  - Fixing them in the greedy order 按贪婪顺序固定它们
  - Propagate fixing and bound changes for each fix 传播变量固定和收紧界值
    - Without it, it is almost impossible to find a feasible solution
    - Solve the remaining LP model, if there are continuous variables
- LP based greedy heuristics 基于LP的贪婪算法
  - It is often more effective to use relaxation solution to sort variables 用松弛解对变量排序通常更有效

Non-LP Based Heuristics 基于非LP启发式算法

- They can find integer solutions quickly 有可能可以快速找到整数解
- The quality of the solutions is often very poor 解的质量通常很差
- One poor solution is good enough, poor solutions often won’t help overall optimization 一个差的解就足够了, 差的解往往无法帮助整体优化
- Multi cores and difficulty to parallelize the root node are the reasons to pay some attention to non-LP based heuristics 多核电脑和难以对根节点并行化是关注基于非LP的启发式算法的原因

LP Based Heuristics 基于LP启发式算法

- Rounding 取整法
  - Solve LP relaxation, round the solution values to nearest integer values 解LP松弛问题，将解值四舍五入到最接近的整数值
  - 0-1 knapsack example (same example) 0-1背包例子(相同例子)

    Max 10 u + 8 v + 11 x + 7 y + 5 z

    s.t. 3 u + 4 v + 6 x + 4 y + 3 z ≤ 14

    u, v, x, y, z are binary variables

    Optimal LP relaxation solution (u, v, x, y, z) = (1, 1, 1, ¼, 0)

    Rounded solution (u, v, x, y, z) = (1, 1, 1, 0, 0) is integer feasible with obj. value 29
  - Simple rounding won't work well, especially for models with equalities 简单的取整不会很好，特别是具有等式约束的模型
  - Consider integer values on both sides 考虑两边的整数值
  - Rounding with propagating, fixing variables and tightening bounds取整时需传播变量固定和收紧界值
  - Gurobi has several different versions of rounding heuristics Gurobi有多种不同版本的取整启发式算法
- Most Gurobi heuristics are LP based or need LP relaxation solutions 大多数Gurobi启发式算法都是基于LP或需要LP松弛解
  - LP relaxation solution is very important for heuristics to get high quality solutions! LP松弛解对于启发式算法获取高质量解非常重要!

Reformulation 模型改建

- Example 例子

  Min $3u + 8v + 3w + 2x + 7y + 5z$

  s.t. $3u + 4v - 4w + 8x + 4y + 3z \le 9$

  $5u + 2v + 4x + 7y + 9z = 15$

  u, v, x, y, z are non negative integer variables, w is a binary variable
- Zero-objective heuristic 去目标启发式
  - Remove the objective and solve it as a feasible problem 删除目标并将其为可行问题去解
  - Hope that presolve can have more reductions and the resulting presolved model is easier to solve 希望预预优化可以使模型变的更小，并且最终的预优化模型更容易解
  - For this example, the reformulated model is 对于这个例子，重新改建的模型是

  Min 0

  s.t. $3u + 4v - 4w + 8x + 4y + 3z \le 9$

  $5u + 2v + 4x + 7y + 9z = 15$

  u, v, x, y, z are non negative integer variables, w is a binary variable
  - Variables x and v are parallel, x can be fixed to 0
  - Variable w can be fixed to 1, which will only help the feasibility of the first constraint

Reformulation 模型改建

- Minimum relaxation heuristic 最小松弛启发式
  - For each inequality, add one penalty variable for the constraint violation 对于每个不等式，加一个违反约束的惩罚变量
  - For each equality, add two penalty variables for the two directions of the violation对于每个等式，加两个违反约束的惩罚变量, 两个方向各一个
  - Then minimize the sum of violations 然后对违约总和求最小化
  - If the optimal solution has the sum = 0, then we find a feasible solution; otherwise the original model is infeasible 如果最优总和为0, 那我们找到一个可行解; 否则原始模型是不可行的
  - For this example, the reformulated model is 对于这个例子，重新改建的模型是

$$
\begin{aligned}
\text{Min} \quad & r + s + t \\
\text{s.t.} \quad & 3u + 4v - 4w + 8x + 4y + 3z - r \le 9 \\
& 5u + 2v + 4x + 7y + 9z + s - t = 15
\end{aligned}
$$

u, v, x, y, z are non negative integer variables

w is a binary variable

r, s, t are non negative continuous variables

RINS

- Relaxation induced neighborhood search (RINS) 松弛诱导邻域搜索
- Given the incumbent (the best integer solution found so far) and the current fractional solution of the node relaxation 给定现任整数解和节点松弛的当前分数解
- Fix a variable if its incumbent value and its relaxation value agrees 如果变量的整数解值和松弛解值一致，则固定变量
- Solve the partially fixed model as a subMIP 将部分固定的模型作为子MIP去解
- It is an improvement heuristics 这是一种改进型的启发式算法
- It is the our most effective heuristic 这是我们最有效的启发式算法

SubMIP and Recursive Solve 子MIP和递归

- Many Gurobi heuristics will 许多Gurobi启发式算法会
  - Have a target to fix some percentage of variables, say 80% 设一个目标来固定一定比例的变量，比如80%
  - Fix one variable and then propagate 固定一个变量然后传播
  - Repeat fixing and propagating until the target is reached or it becomes infeasible 重复固定和传播，直到达到目标或它变得不可行
  - Solve it as a subMIP 将其为子MIP去解
  - In the subMIP, it will call the same heuristics, so recursively 在子MIP中,它将以递归方式调用相同的启发式算法
  - It often works well and finds feasible solutions quickly 它通常非常有效, 可迅速找到可行解

Feasibility Pump Heuristic泵式缩减启发式

- Fischetti, Glover and Lodi, 2004
- Solve the relaxation and round the solution解松弛问题并舍入到整数解
- Replace the objective to minimize the distance to the rounded solution (quadratic)目标换成到舍入整数解距离最小 (二次）
- Use L1 norm (sum $|x_j – x_j^*|$), where x* is the rounded solution (linear)使用L1范数(sum | xj - xj * |)，其中x *是取整解（线性）
  - If a binary variable $x_j = 0.3$, then $x_j^* = 0$, then the objective part for x_j is $| x_j – 0| = x_j$, i.e. obj. coefficient is 1
  - If a binary variable $x_j = 0.7$, then obj. coefficient will be -1
- Solve the modified LP and repeat 解修改后的LP并重复
- Until it hits some limit or the relaxation solution is integer feasible直到它达到一定限值或松弛解是整数可行的
- Setting the limit to e.g. 10, i.e. solving the LP 10 times is expensive and it usually won’t be lucky例如将限值设为10, 则需解LP10次, 很化时, 通常很难运气好

Pump Reduce泵式缩减

- Motivated by feasibility pump heuristic 受泵式缩减启发式算法的启发
- Observation 观察
  - Most models are dual degenerate, i.e. relaxation has alternative optimal solutions大多数模型对偶退化,即松弛问题有多个的最优解
- Goal 目标
  - A relaxation solution with less fractional integer variables 有较少整数变量取分数值的松弛解
    - Possible zero fractional integer variables, but not the goal, so it isn't heuristic 可能没有分数整数变量，但不是目标，所以它不是启发式的
    - Such relaxation solution helps heuristics and b&b significantly to find integer feasible solutions 这样的松弛解很显著地帮助启发式和b＆b找到可行的整数解
- Steps 步骤
  - Solve the relaxation and fix all variables with nonzero reduced costs, making sure to stay in the optimal space 解松弛问题, 固定非零递减成本的所有变量, 确保保持在最优空间
  - Round the relaxation solution, replace the objective with L1 norm distance to the rounded solution舍入松弛解, 目标换成到舍入整数解距离最小 (L1 范数)
  - Solve the modified LP, round and repeat解修改后的LP并重复
  - Until it hits some limit or the number of fractional integer variables doesn't go down直到它达到一定限值或取分数值的整数变量的数量不下降

GUROBI Heuristic Parameters

Gurobi 启发式参数

Heuristic Parameters 启发式参数

- Main MIP parameter 主要MIP参数, MIPFocus
- Main heuristic parameter 主要启发式参数, Heuristics
- Individual heuristic parameters 个别启发式参数
- Other parameters affecting feasible solutions 影响可行解的其他参数

MIPFocus

- Define high-level solution strategy 定义解高层策略
- Default 默认, balance between finding new feasible solutions and proving that the current solution is optimal. 在找新的可行解和证明最优性之间取得平衡
- = 1, more interested in finding feasible solutions quickly 更注重找到可行解
- = 2, more attention on proving optimality 更注重证明最优性
- = 3, focus on the objective bound 更注重目标界值

Heuristics

- Main heuristic parameter主要启发式参数
- The parameter value is roughly the fraction of time that we will spend on heuristics 参数值大致是我们在启发式上花费时间的部分值
- Default value默认值= 0.05
- > 0.05, more aggressive, 1 most aggressive
- < 0.05, less aggressive, 0 no heuristics

Individual heuristic parameters个别启发式参数

- Pump reduce 泵式缩减 (or degenerate simplex moves), Degenmoves
- Feasibility pump heuristic泵式缩减启发式, PumpPasses
- Improvement heuristic parameters改进型的启发式参数
  - ImproveStartGap
  - ImproveStartNodes
  - ImproveStartTime (warning: not deterministic)
- Minimum relaxation heuristic最小松弛启发式, MinRelNodes
- RINS heuristic RINS启发式, RINS
- Zero objective heuristic去目标启发式, ZeroObjNodes

Other Parameters Affecting Heuristics
影响启发式的其他参数

- Nodes explored by sub-MIP heuristics, SubMIPNodes
- Branch direction preference, BranchDir
  - Setting the value to 1 may help MIP diving to find a feasible solution more quickly
- Tuning criterion, TuneCriterion
  - = 2 objective value, i.e. focusing more on finding good feasible solutions

User Input for GUROBI Heuristics

启发式的用户输入功能

MIP Start / Multiple MIP Starts MIP起始值/多个MIP起始值

- User can provide a MIP start or multiple MIP starts (new in 8.0) 用户可以提供一个或多个MIP起始值(多个为8.0的新功能）
- A good MIP start, even a partial solution often can produce a good feasible solution instantly 良好的MIP起始值，即使是部分解，也可以立即产生好的可行解
- Useful when you have multiple partial solutions 有多个部分解可能会很有用
  - MIP solver will try to complete them, and will store the ones it finds
- For distributed MIP, MIP starts will be evaluated on different machines 对于分布式MIP，将在不同的机器上评估MIP起始值

Variable Hints MIP提示

- Provide hints to the solver about which variable should take which value 向优化器提示哪个变量应采用哪个值
- Guides heuristics and branching 指导启发式和分支
- VarHintVal attribute 属性
  - Specifies a value for a variable 指定变量的值
- VarHintPri attribute 属性
  - Specifies a level of confidence in this particular variable value 指定此特定变量值的置信度
- Comparison to MIP start 与MIP起始值比较
  - MIP start is used to provide an initial feasible solution to the solver MIP起始值用于为优化器提供初始可行解
    - Is evaluated prior to starting the solution process
    - Provides incumbent if feasible
    - Does not influence solution process if it is not feasible
  - Variable Hints guide the search 变量提示指导搜索
    - High quality hints should lead to a high quality solution quickly
      - Either through heuristics or through branching
    - Affects the whole solution process

Partition Heuristic分区启发式算法

- User-specified local improvement heuristic 用户指定的局部改进型启发式算法
- RINS is our most effective heuristic RINS是我们最有效的启发式算法
- It is a *sub-MIP* heuristic 这是一个子MIP启发式算法
  - Fix a subset of the variables to incumbent values 将变量的子集固定为现任整数解的值
  - Solve the resulting MIP (recursively) 解生成的MIP(递归）
    - Reoptimizes over just that portion of the problem
- Sub-MIP heuristics extremely effective in general 子MIP启发式算法一般非常有效
- How to choose the sub-problem to reoptimize? 如何选择子问题进行重新优化？
  - RINS chooses automatically RINS自动选择
  - This feature allows user to make the choice 此功能允许用户做出选择
    - Example sub-problems:
      - All decisions related to a single time period
      - All decisions related to a single machine
      - All decisions related to physical sub-regions (e.g., Western US, Eastern US, etc.)

MIP Heuristic Callback 启发式回调

- Motivations
  - Our MIP solver is mostly a black box solver 我们的MIP求解器主要是黑盒求解器
    - We try to recognize some common structures, but very limited
  - Users know the structure of their model 用户知道他们模型的结构
  - Relaxation solutions help heuristics a lot 松弛解对启发式算法很有帮助
  - Knowledge of problem structure and the relaxation solutions often mean fast good feasible solutions 对问题结构和松弛解的了解通常意味着快速找到可行解
- Heuristic callback
  - At each node, Gurobi will call back 在每个节点，Gurobi都会回调
  - Users can query the relaxation solution and use it to guide their heuristics 用户可以查询松弛解并使用它来指导他们的启发式算法
  - Users can provide a full or partial solution vector to Gurobi through callback 用户可以通过回调向Gurobi提供完整或部分解
  - If it is partial, Gurobi will try to complete it 如果它是部分的，Gurobi将尝试完成它

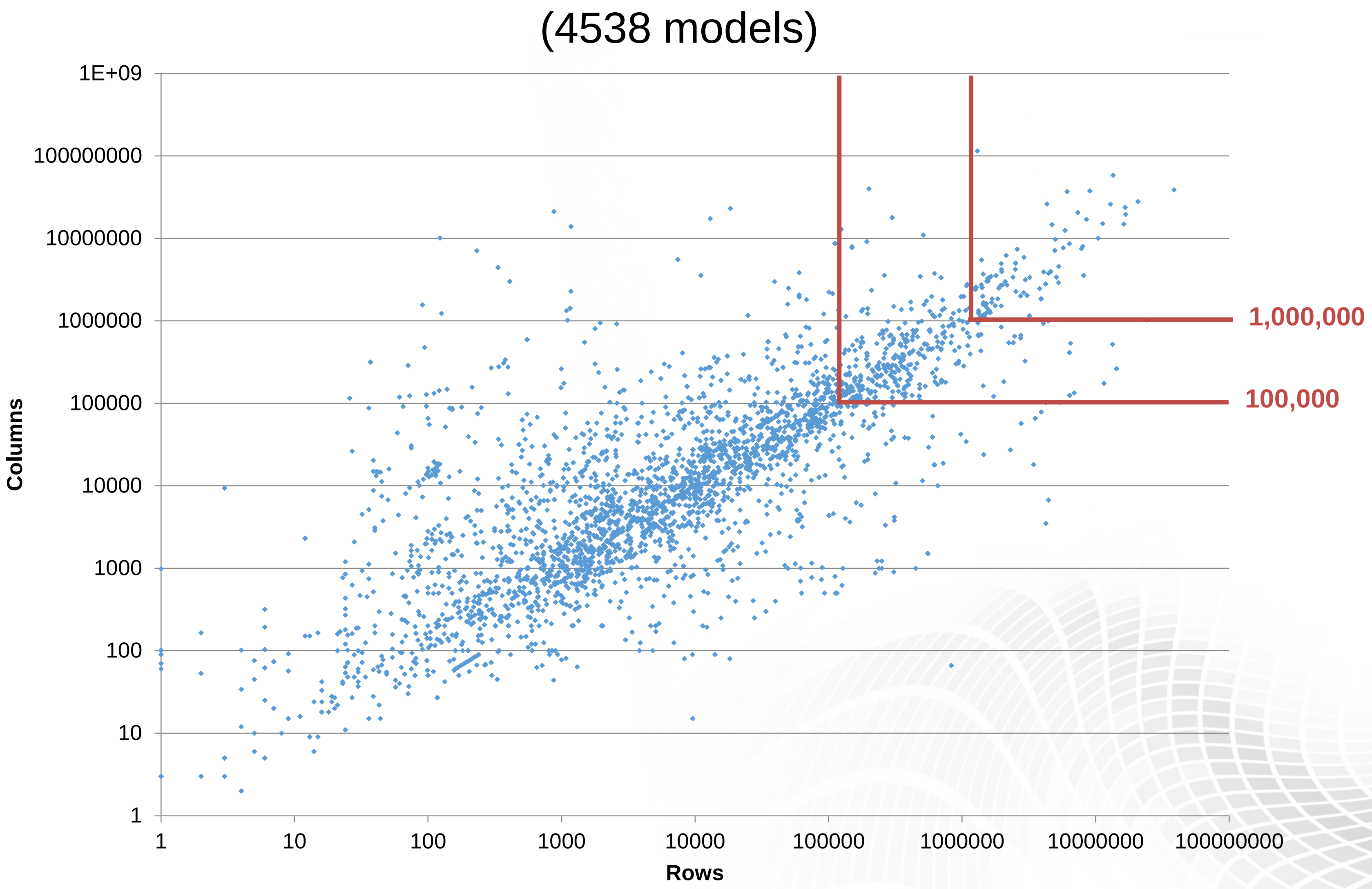
What to do with too big/hard models

如何处理太大/太难的模型

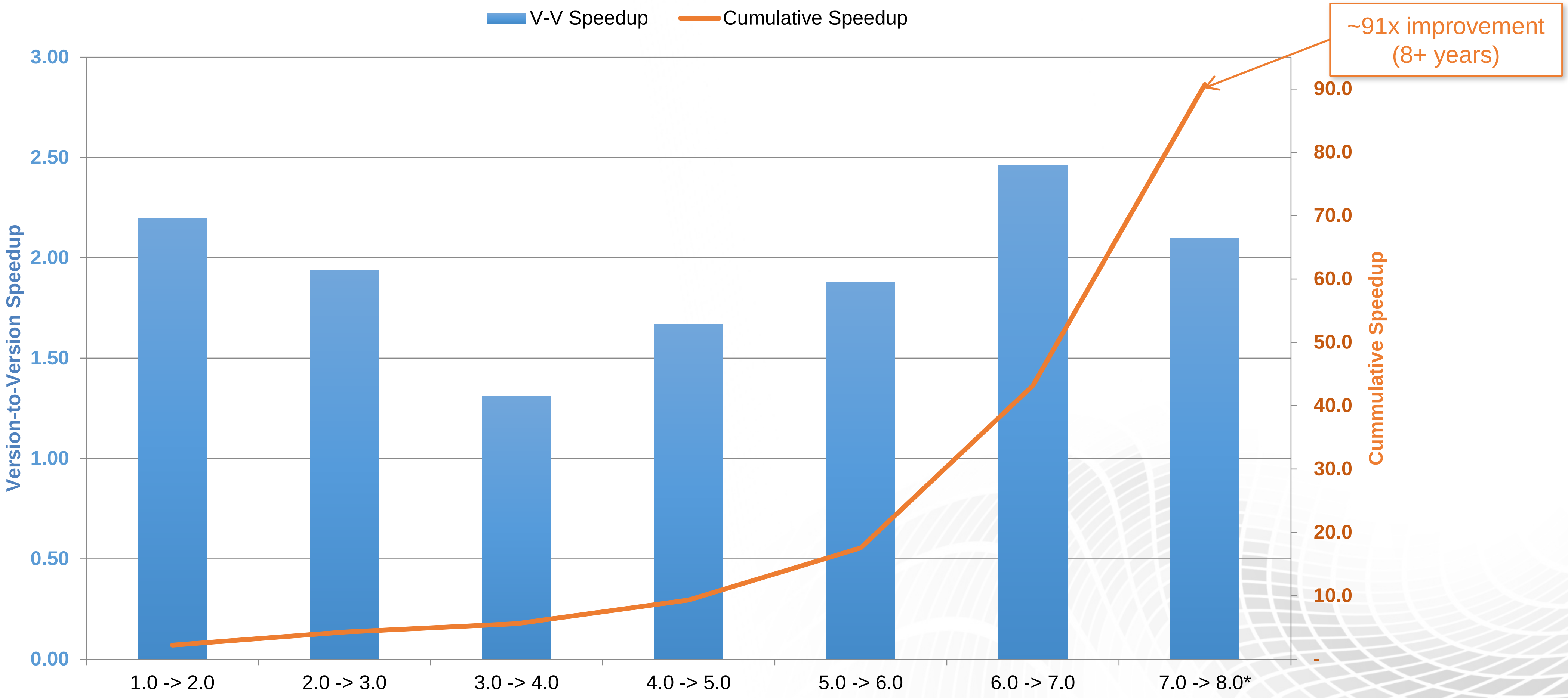
Too Big/Hard Models, Really?太大/太难的模型, 真的吗？

- “My model is too big or too hard, I have no choice but heuristic”, really? “我的模型太大或太难，我别无选择,只有启发式算法”,真的吗？
- Old MIP experiences don’t count 旧的MIP 经验不算数
  - At the end of 80’s and earlier 90’s, people in electrical power industry concluded that MIP was a nice tool, which couldn’t solve real unit commitment model
  - Close to 2000, people revisited MIP technology and people now solve the unit commitment model routinely
  - The similar stories happened more and more
- I tried open source solvers, they are hopeless 我试过开源优化器，没有希望解我的问题
  - We have a lot of users, who send us their models, since the open solver they used couldn’t find a feasible solution in hours. Gurobi often solved the models in less than one second
  - All open source solvers are way behind the state of art commercial solvers
- Gurobi users often solve their MIP models with millions of variables/constraints Gurobi用户经常解有数百万个变量/约束的MIP模型
  - Our customer model sets have a lot of such models, many of them we can solve or find good solutions within 10% MIPGap.

Gurobi MIP Library 模型集



MIP速度不断提高, 主要版本每次提高几乎两倍



True Very Big/Hard Models 真的太大/太难的模型

- Have you tried to solve the relaxation? 你试过解松弛问题吗？
  - LP relaxation is polynomial-time solvable LP松弛问题是多项式时间可解的
  - Gurobi has solved LP models with 100M+ variables/constraints Gurobi解过许多超过几亿个变量/约束的LP模型
  - Relaxation solution is often very useful for heuristics松弛解对启发式算法很有帮助
  - The objective value of the relaxation solution provides the bound, without it, it is hard to know how good a heuristic solution is 松弛解提供目标界值，没有它，很难知道启发式解有多好

True Very Big/Hard Models 真的太大/太难的模型

- Have you tried to reduce the models?你试过减小模型吗?
  - Aggregate汇总
    - Daily schedule -> weekly schedule日计划->周计划
  - Decompose big model into smaller pieces 将大模型分解为较小的部分
    - World -> America, Europe and Asia 世界 - >美洲,欧洲和亚洲
  - Local improvement 局部改进
    - Use heuristic to generate an initial solution 使用启发式算法生成初始解
    - Use MIP to reoptimize over a portion of the model, like RINS 使用MIP重新优化模型的一部分，如RINS
  - No lower bound, but often produces very high quality global solutions 没有下限，但通常会产生非常高质量的整体解

True Very Big/Hard Models真的太大/太难的模型

- Successful stories to combine optimization and heuristics融合优化和启发式的成功案例
  - MIP based heuristics 基于MIP的启发式算法
    - Rolling horizon heuristics 滚动时段启发式算法
      - Relax integrality of future periods
      - May aggregate future time periods
      - Solve smaller LP/MIP
      - Air taxi and mining
    - Local search heuristics 局部搜索启发式算法
      - In group of periods, machines etc, solve smaller LP/MIP
      - Lenstra et al., local search in combinatorial optimization
  - Lin-Kernighan heuristic for TSP 货郎担问题的启发式算法
    - Solve relaxation and use reduced costs to guide
  - Etc.

Always Try MIP
总是试试MIP

Shabbir Ahmed
@Shabbir0Ahmed

Follow

Richard Karp quotes a colleague "Always try integer programming, it might work"

8:27 AM - 13 Mar 2017

Conclusion
结论

- Always try Gurobi, it should be better than pure heuristics

  试试Gurobi，它应该比纯启发式更好!

Thank you – Questions?

谢谢，请提问题