Gurobi 11.0

Every Solution, Globally Optimized

Gurobi 11.0 新亮点和技术创新

顾宗浩 博士，Gurobi CTO 和联合创始人

GUROBI OPTIMIZATION

主讲人介绍

- **顾宗浩 博士**
- Gurobi 首席技术官 (CTO) 和联合创始人
- 上海同济大学机械工程学士和工业管理硕士，佐治亚理工学院工业工程博士
- 数学规划理论和实践领域全球最顶尖的专家之一

讲座大纲

LP 性能提升

MIP 性能提升

全局 MINLP求解器

API 接口改进: Java & Gurobipy

Q&A

GUROBI OPTIMIZATION

LP 算法的性能提升

© 2023 Gurobi Optimization, LLC. Confidential, All Rights Reserved | 4

性能提升汇总

Gurobi 10.0 对比. Gurobi 11.0 – 连续凸模型

类型	整体提升	>100秒复杂模型
LP (concurrent)	0.7%	1.2%
LP (barrier)	2.0%	4.1%
LP (primal simplex)	3.1%	9.3%
LP (dual simplex)	0.3%	1.9%
QP	1.3%	—*
SOCP	40.6%**	—*

Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v5 @ 3.50GHz
4 cores, 8 hyper-threads
32 GB RAM

* Too few hard QP and SOCP models to measure performance
** Includes performance bug fix for dense cone handling

GUROBI OPTIMIZATION

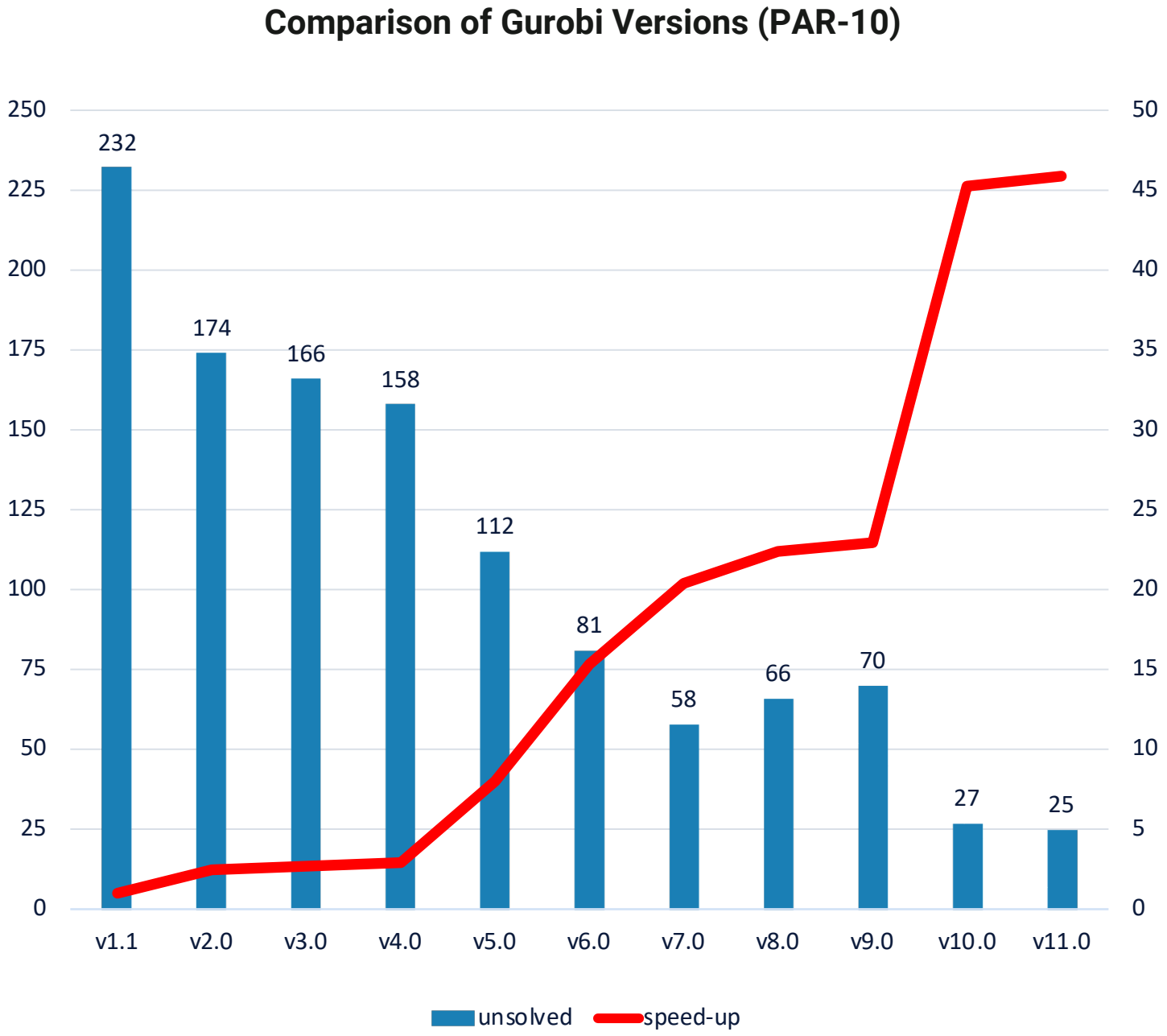
LP
性能演变

默认设置:

- Gurobi 1 – 4: dual simplex
- Gurobi 5+: concurrent LP

Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v5 @ 3.50GHz
4 cores, 8 hyper-threads
32 GB RAM

Test set has 2574 models:
- 225 discarded due to inconsistent answers
- 77 discarded that none of the versions can solve
- speed-up measured on >100s bracket: 596 models



SolutionTarget 参数

- 首先出现在 Gurobi 10.0.0，做为未公开参数
- 在 Gurobi 11.0 中完整记录并在所有API中实现

- `SolutionTarget`:
  - `-1`: 自动选择 (等同于 SolutionTarget=0)
  - `0`: 获得基最优解
  - `1`: 获得最优解（不一定是基最优）

- 可以获得更好的求解速度，特定步骤可以跳过:
  - 内点算法之后的交叉转换Crossover
  - 对称折叠消减之后展开至基解

SolutionTarget 参数

性能改进

- Gurobi 10.0:
  - `SolutionTarget=1`基本上等同于 `Method=2 Crossover=0`
- Gurobi 11.0:
  - 支持并发LP中包含没有crossover的内点算法
  - Gradient Boosted Tree 算法决定采用单独内点算法还是并发LP算法
- 相比于 Gurobi 10.0 的性能提升:
  - 整体提升 23% (>100秒模型提升 46%)*

* concurrent LP with SolutionTarget=1, 4 core machine

并发LP算法参数

Gurobi 10.0

- `Method`
  - `-1`: automatic
  - `0`: primal simplex
  - `1`: dual simplex
  - `2`: barrier
  - `3`: non-deterministic concurrent LP
  - `4`: deterministic concurrent LP
  - `5`: deterministic concurrent simplex

Gurobi 11.0

- `Method`
  - `-1`: automatic
  - `0`: primal simplex
  - `1`: dual simplex
  - `2`: barrier
  - `3`: non-deterministic concurrent LP
  - `4`: deterministic concurrent LP
  - `5`: deterministic concurrent simplex (deprecated)
- ConcurrentMethod
  - `-1`: automatic
  - `0`: barrier/dual/primal
  - `1`: barrier/dual
  - `2`: barrier/primal
  - `3`: dual/primal

GUROBI OPTIMIZATION

MIP 算法的性能提升

© 2023 Gurobi Optimization, LLC. Confidential, All Rights Reserved | 10

性能提升汇总

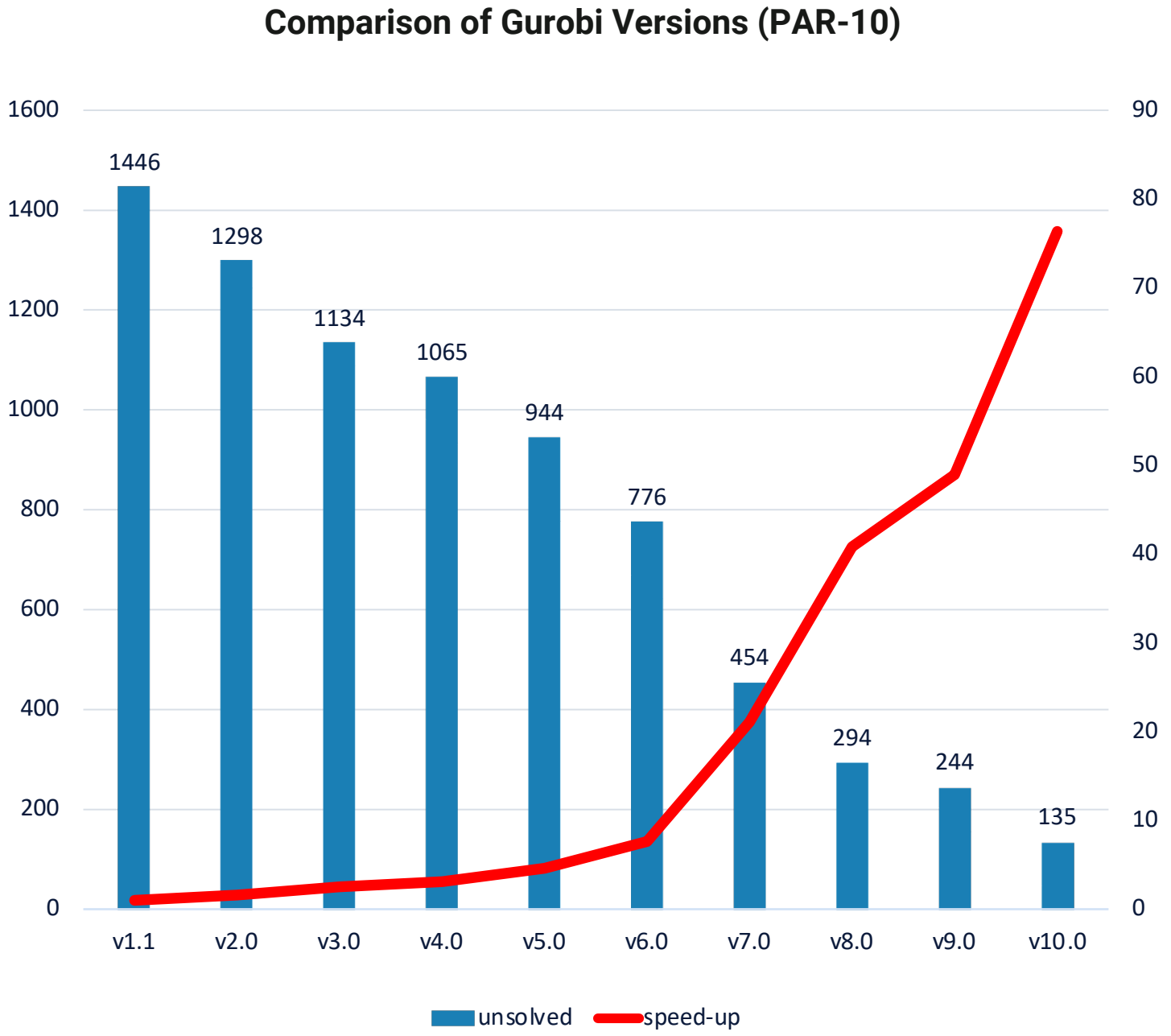
Gurobi 10.0 对比 Gurobi 11.0 – 混合整数凸模型

类型	整体提升	>100秒复杂模型
MILP	8.6%	12.4%
MIQP	12.8%	22.8%
MIQCP	9.2%	18.2%

Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v5 @ 3.50GHz
4 cores, 8 hyper-threads
32 GB RAM

GUROBI OPTIMIZATION

MILP
性能演变



Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v5 @ 3.50GHz
4 cores, 8 hyper-threads
32 GB RAM

Test set has 7102 models:
- 696 discarded due to inconsistent answers
- 1871 discarded that none of the versions can solve
- speed-up measured on >100s bracket: 2623 models

MIP 性能提升方法

众多微小改进累积发挥作用 (1/2)

GUROBI OPTIMIZATION

- 预优化:
  - 改进探测中的传播算法 Improved propagation in probing (0.6%)
  - 改进探测中的提升次序 Improved lifting sequence in probing (0.7%)
  - 改进探测中的工作跟踪 Improved work tracking in probing (0.3%)
  - 聚类(clique)表格中删除重复记录，采集替代记录 Discard duplicates from clique table, collect substitutions (0.7%)
  - 更好处理大规模模型缩减算法 Better handling of significant model size reductions (1.3%)
- 节点预优化:
  - 区域传播算法中的工作极限调整 Work limit adjustment in domain propagation (0.3%)
  - 区域传播算法中更好处理数值问题 Better numerics in domain propagation (0.3%)
  - 更早从递减成本固定中更新全局上下界 Earlier updates of global bounds from reduced cost fixing (2.1%)
- 分支:
  - 为分支变量选择计算 GMI 切平面分数 GMI cut scores for branching variable selection(0.3%)
  - 为分支变量选择计算已传播伪成本分数 Propagated pseudo cost scores for branching variable selection (1.0%)

MIP 性能提升方法

众多微小改进累积发挥作用 (2/2)

- 切平面:
  - 分离高级别背包覆盖切平面 Separate higher rank knapsack cover cuts (1.0%)
  - 改进聚类切平面分离算法 Improved clique cut separation (0.6%)
  - 更激进的约束分解切平面 Aggressive constraint disaggregation cuts (0.8%)
  - 改进根节点切平面并行循环算法 Improvements in parallel root cut loops (0.4%)
  - 更激进的子MIP切平面算法 More aggressive sub-MIP cuts (1.5%)
  - 混合路径切平面算法 Mixing Path Cuts (1.2%)
- 对称性:
  - 更早在根节点探查对称问题 Detect and exploit symmetry earlier at root node (0.9%)
  - 弱对称性切平面Weak symmetry cuts (0.9%)
- 其他:
  - 改进冲突约束的选择算法 Improved conflict constraint selection (0.8%)
  - 改进排序算法 Improved sorting methods (0.4%)

混合路径切平面

GUROBI OPTIMIZATION

- 文献:
  - O. Gunluk, Y. Pochet: Mixing mixed-integer inequalities.
    Math. Program. 90, 429–457 (2001). https://doi.org/10.1007/PL00011430
  - P. Christophel: Separation algorithms for cutting planes based on mixed integer row relaxations: implementation and evaluation in the context of mixed integer programming solver software. (PhD thesis)
    University of Paderborn, 2009, pp. 1-222
- 融入我们的 MIR 聚合过程, 使用 Christophel 的 “U-cut procedure”
- `MixingCuts` 参数，取值 -1, 0, 1, 2
- 性能改进: 整体提升 0.5%, >100秒模型提升 1.2%

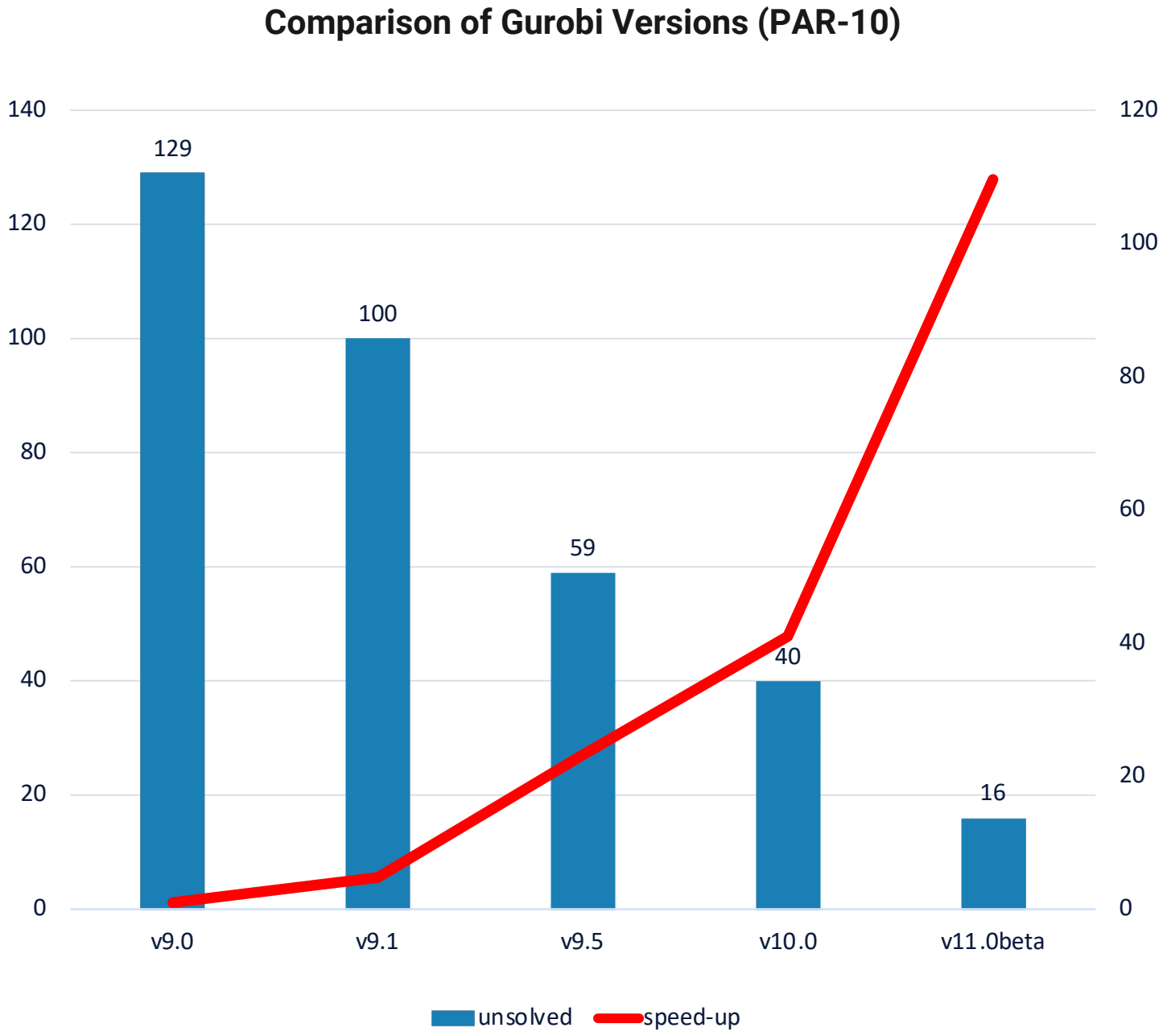
GUROBI OPTIMIZATION

非线性算法的性能提升

© 2023 Gurobi Optimization, LLC. Confidential, All Rights Reserved | 16

GUROBI OPTIMIZATION

非凸 MIQCP
性能演变



Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v5 @ 3.50GHz
4 cores, 8 hyper-threads
32 GB RAM

Test set has 874 models:
- 40 discarded due to inconsistent answers
- 278 discarded that none of the versions can solve
- speed-up measured on >100s bracket: 237 models

性能提升汇总

Gurobi 10.0 对比 Gurobi 11.0 –非凸模型

类型	整体提升	>100秒复杂模型
nonconvex MIQCP	2.3x	5.8x

Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v5 @ 3.50GHz
4 cores, 8 hyper-threads
32 GB RAM

二次目标和约束

NonConvex 参数

- `NonConvex`:
  - `-1`: 自动判断
  - `0`: 报错，如果原模型包含非凸目标或者约束 Q 矩阵
  - `1`: 报错，如果预优化之后的模型含有无法线性化的非凸目标或约束 Q 矩阵
  - `2`: 利用双线性变换直接处理非凸目标和约束 Q 矩阵
- Gurobi 10.0 默认值 (-1): 等同于设置 1
- Gurobi 11.0 默认值 (-1): 等同于设置 2

- 这个变化有可能影响用户原程序或者给用户意外！
- 经常，非凸可能是模型或者数据有错误
  - 用户这时候也许想要设置 `NonConvex=1`

非线性约束

- Gurobi 9.0 版本开始，提供了定义非线性函数的接口
  - e^x, a^x
  - $\ln(x), \log_a(x)$
  - $\sin(x), \cos(x), \tan(x)$
  - x^a
  - $ax^3 + bx^2 + cx + d$

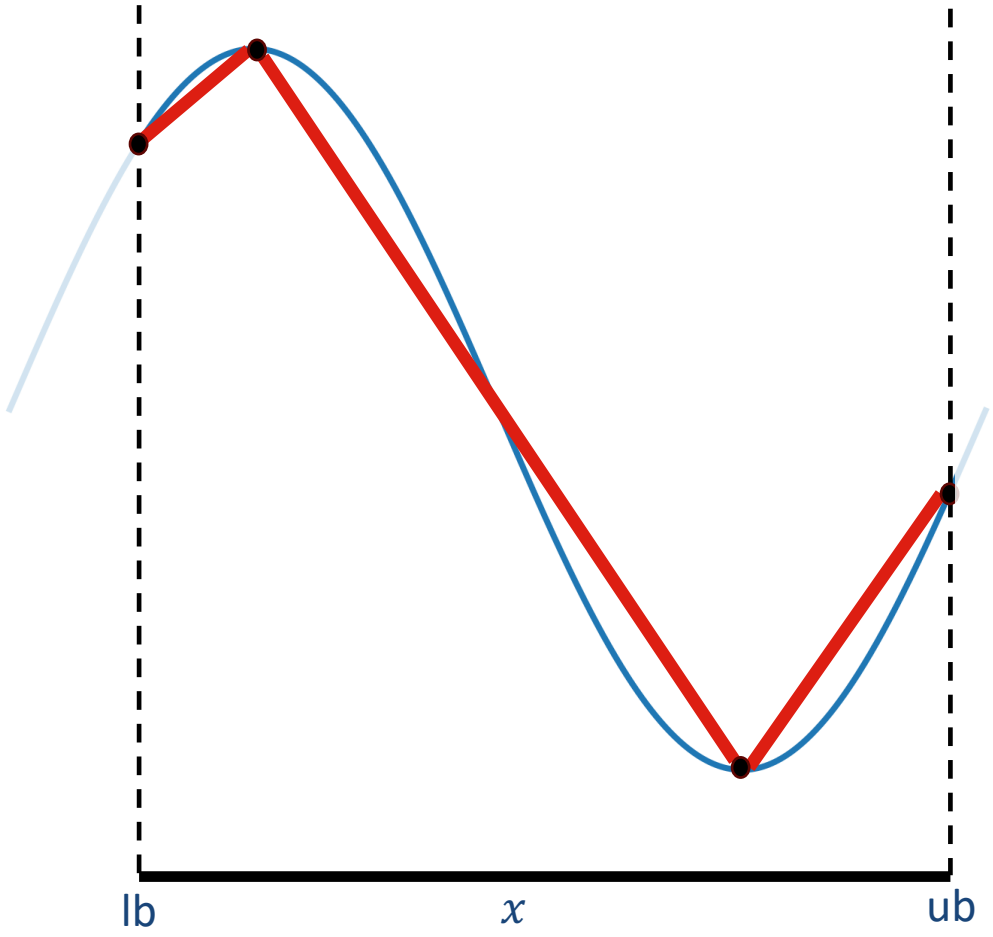
```
addGenConstrExp(), addGenConstrExpA()
addGenConstrLog(), addGenConstrLogA()
addGenConstrSin(), addGenConstrCos(), addGenConstrTan()
addGenConstrPow()
addGenConstrPoly()
```

- Gurobi 9.0 – 10.0:
  - 非线性函数在预优化阶段被替换为分段线性近似函数
- Gurobi 11.0:
  - 支持全局精确处理非线性函数

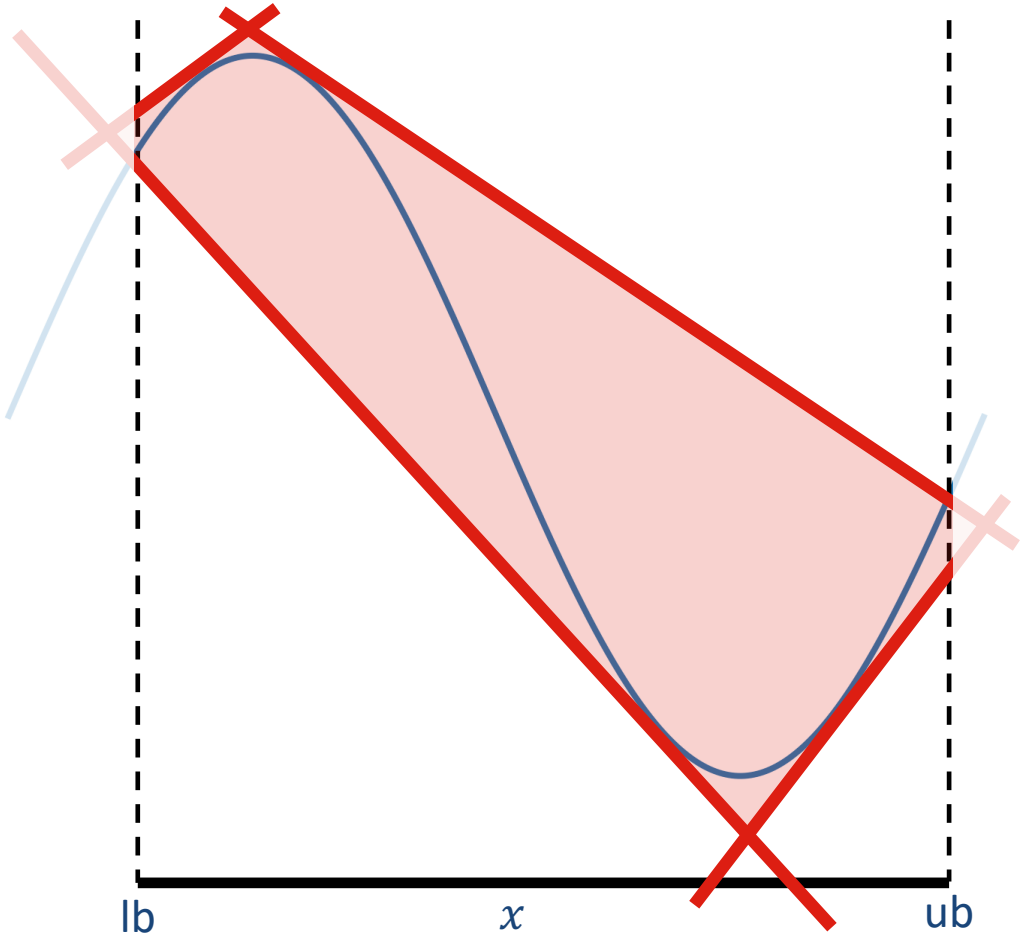
FuncNonlinear 参数和属性

- 已有的控制分段线性近似算法精度的优化参数:
  - `FuncPieces`
  - `FuncPieceError`
  - `FuncPieceLength`
  - `FuncPieceRatio`
- `FuncPieces` 默认值设定为相对误差模式
  - 在Gurobi 10.0中主要用来限制分段的总数
- 新增加的 `FuncNonlinear` 属性用来选择分段线性近似方法和外逼近方法:
  - `-1`: 默认选择由 FuncNonlinear 参数来确定
  - `0`: 采用静态分段线性近似方法
  - `1`: 采用动态外逼近方法
- 新增加的 `FuncNonlinear` 参数控制默认属性值的设定:
  - `0`: 采用静态分段线性近似方法（默认）
  - `1`:采用动态外逼近方法

分段线性近似 对比 外逼近算法

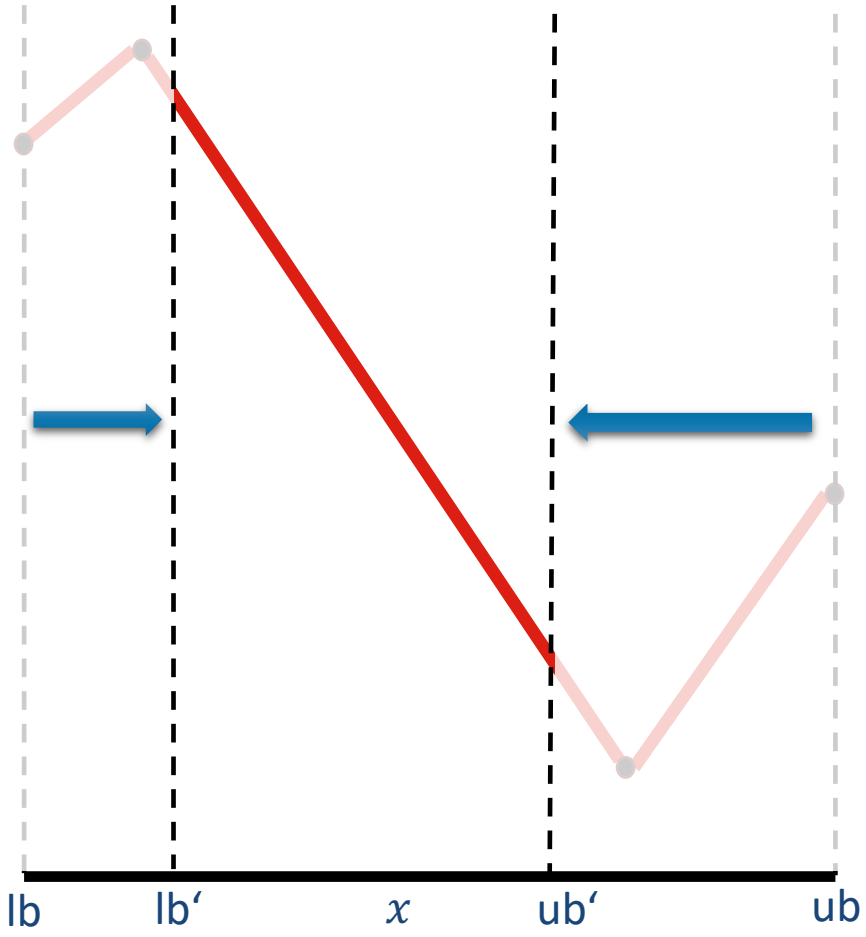


static PWL approximation

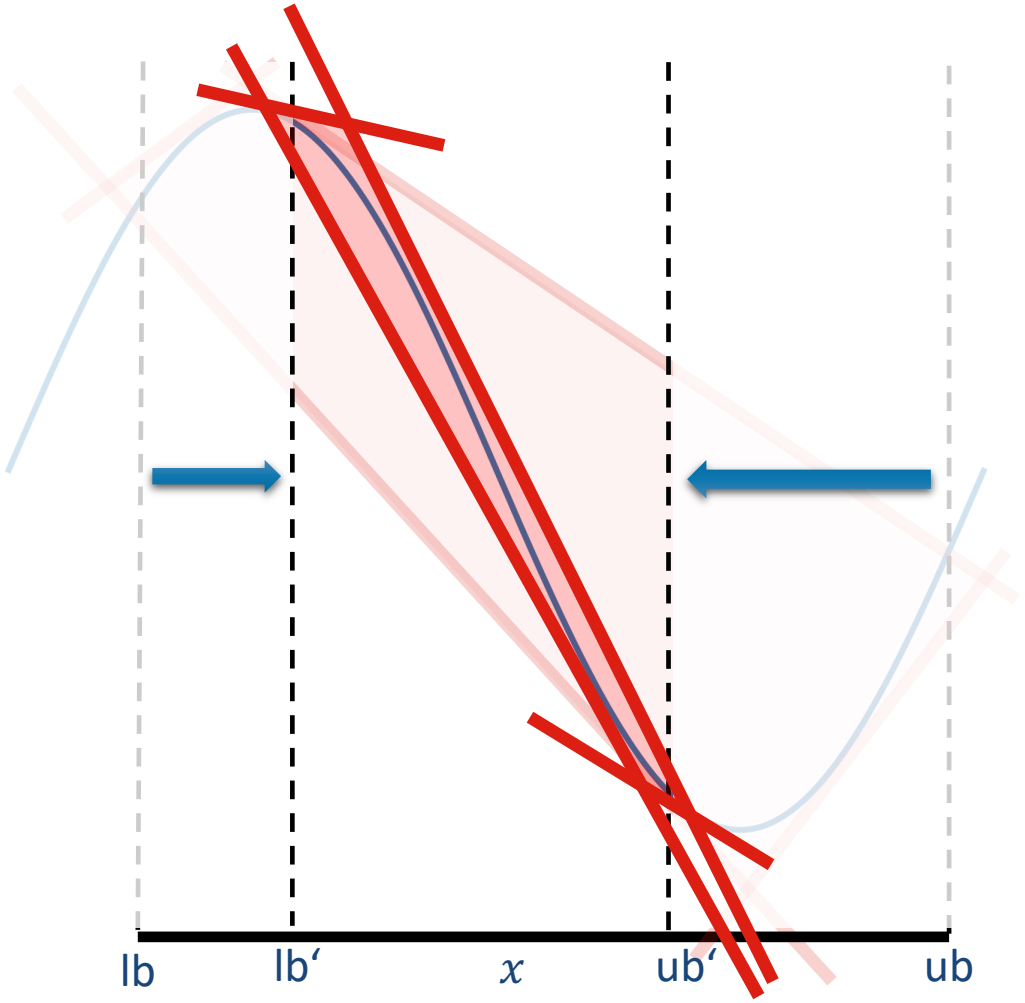


dynamic outer approximation

分段线性近似 对比 外逼近算法



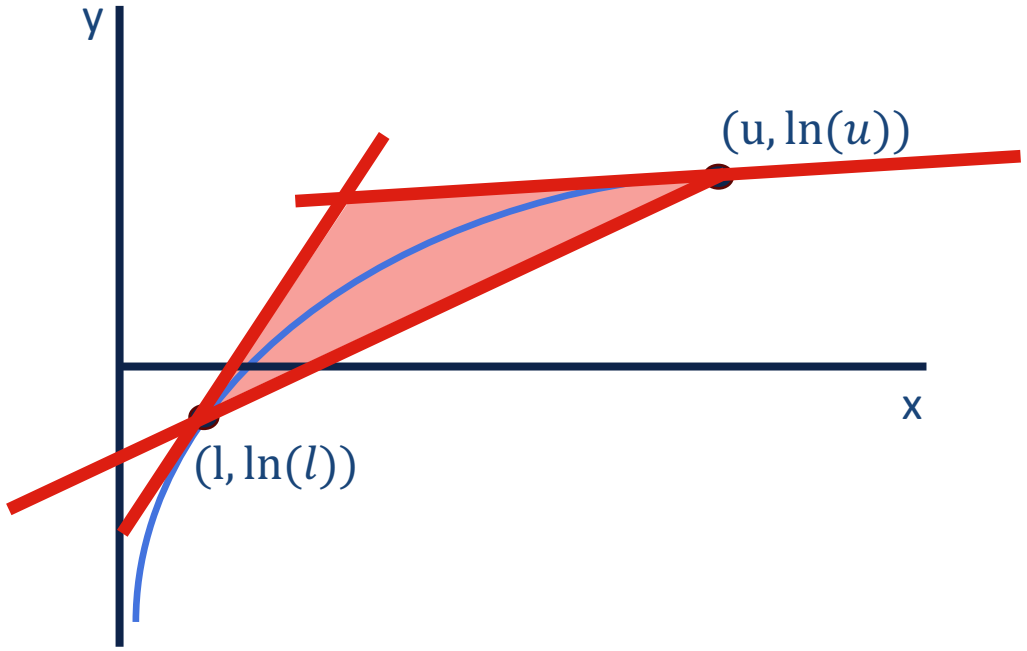
static PWL approximation

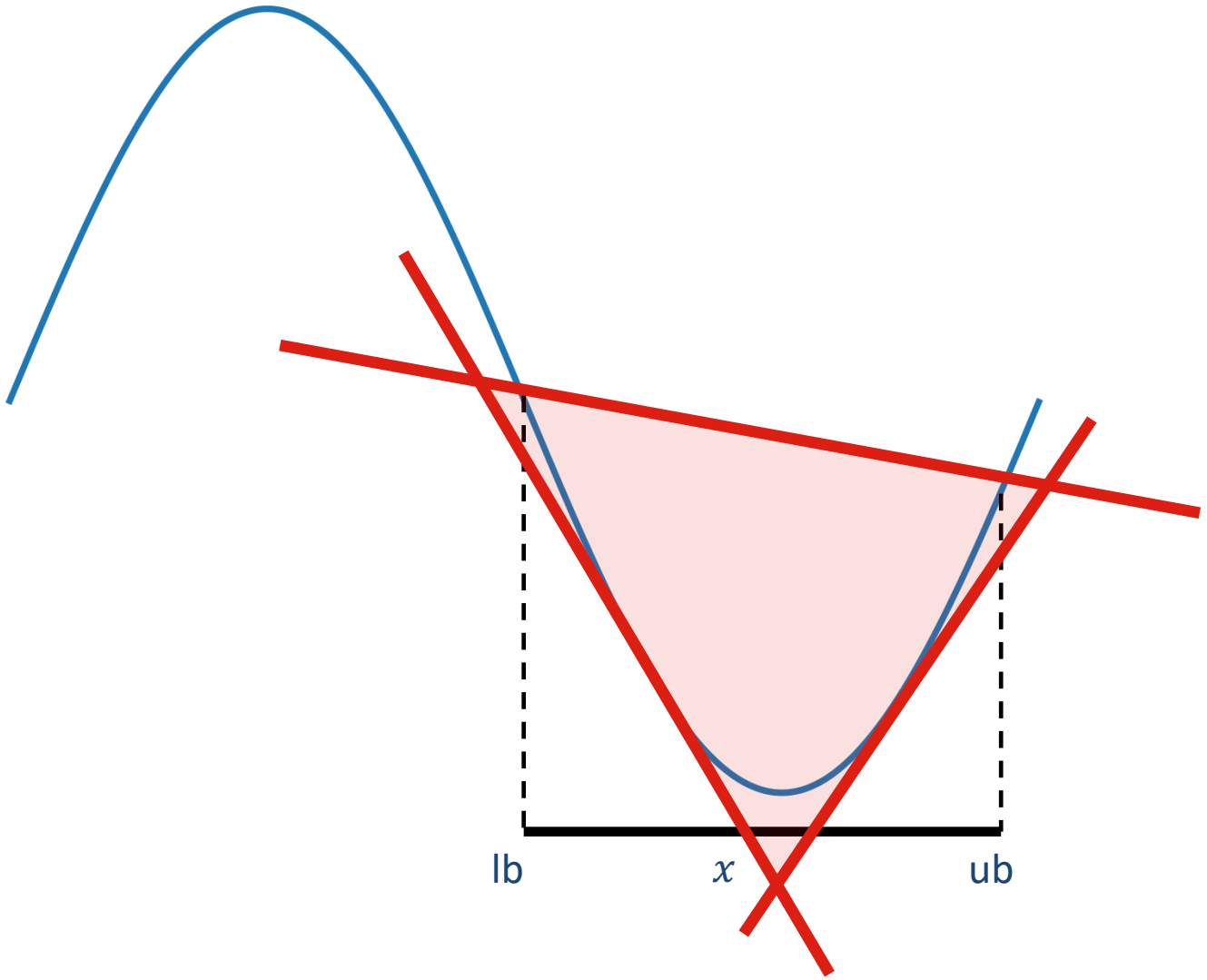


dynamic outer approximation

非线性函数的松弛

- 一些凸包络比其他更容易逼近
  - 举例: $y = \ln(x)$, 一个凹函数, $l \leq x \leq u$
  - 下部包络由通过 $\ln(l)$ 和 $\ln(u)$ 的分割线组成
  - 上部包络由切线组成





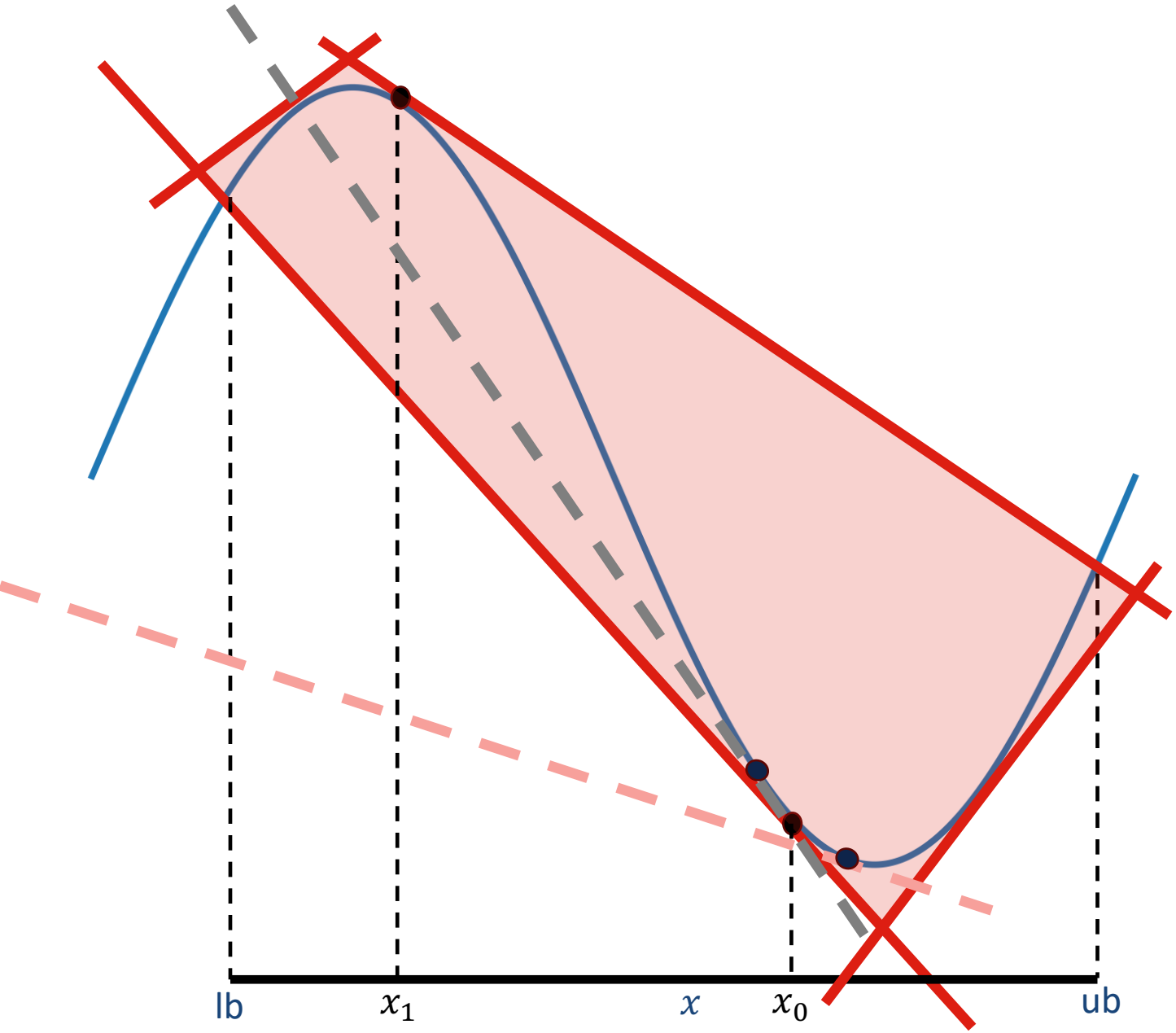
GUROBI OPTIMIZATION

扩展到更复杂的函数

如果 sin 在 x 范围内是凸的...

- 上部包络由通过 $f(\text{lb})$ 和 $f(\text{ub})$ 的分割线组成
- 下部包络由 sin 的切线组成
- 获得的超平面添加到 LP
- 红色区域: $y = f(x)$ 的松弛

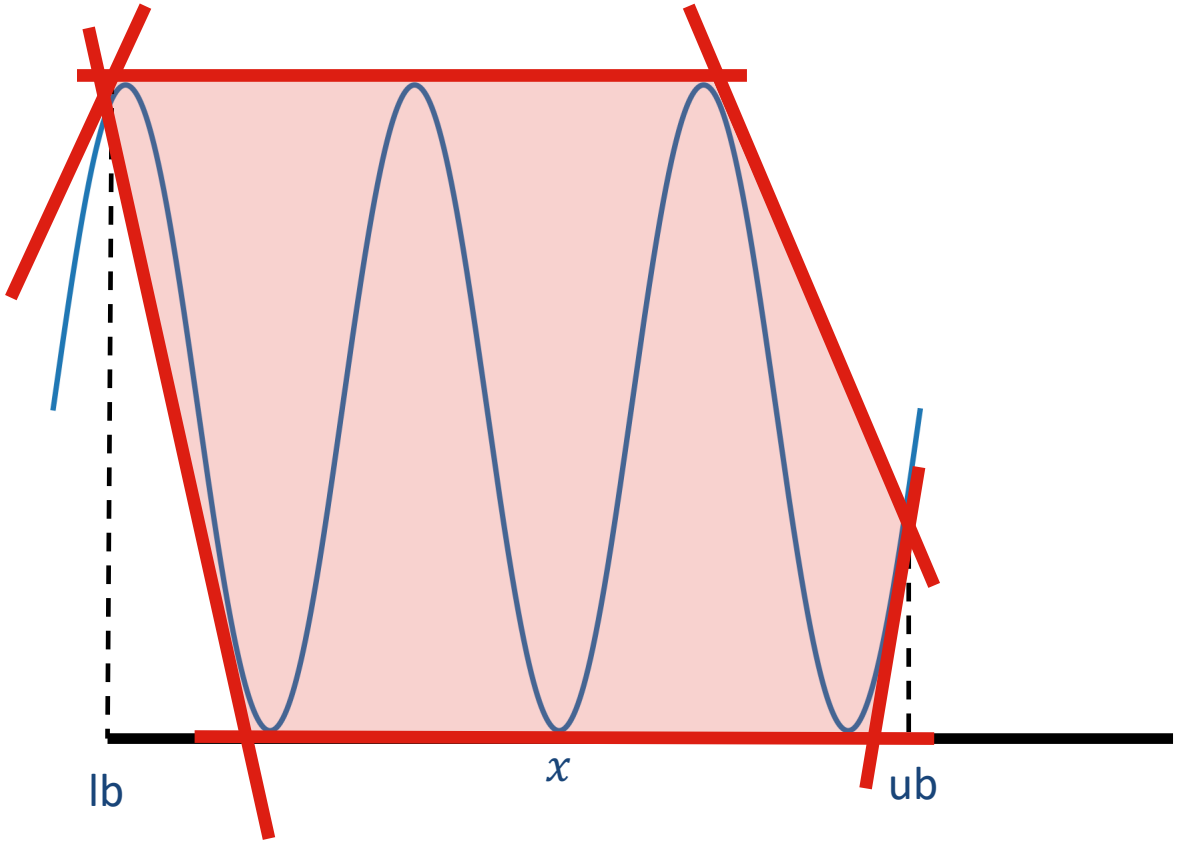
- 与此类似，如果 sin 在 x 范围内是凹的。
- 在不同点添加切线来改进松弛模型



GUROBI OPTIMIZATION

既非凸也非凹

- 如果 sin 在 x 范围内既非凸也非凹
- 下部包络
  - 计算最左边 x_0
    $$\frac{d}{dx}\sin(x) = \frac{\sin(x) - \sin(\text{lb})}{x - \text{lb}}$$
  - 计算后的 x_0 定义一个切线
  - 剩余部分是凸的，利用一些切线
- 上部包络
  - 计算最右边 x_1
    $$\frac{d}{dx}\sin(x) = \frac{\sin(\text{ub}) - \sin(x)}{\text{ub} - x}$$
  - 计算后的 x_1 定义一个切线
  - 剩余部分是凹的，利用一些切线

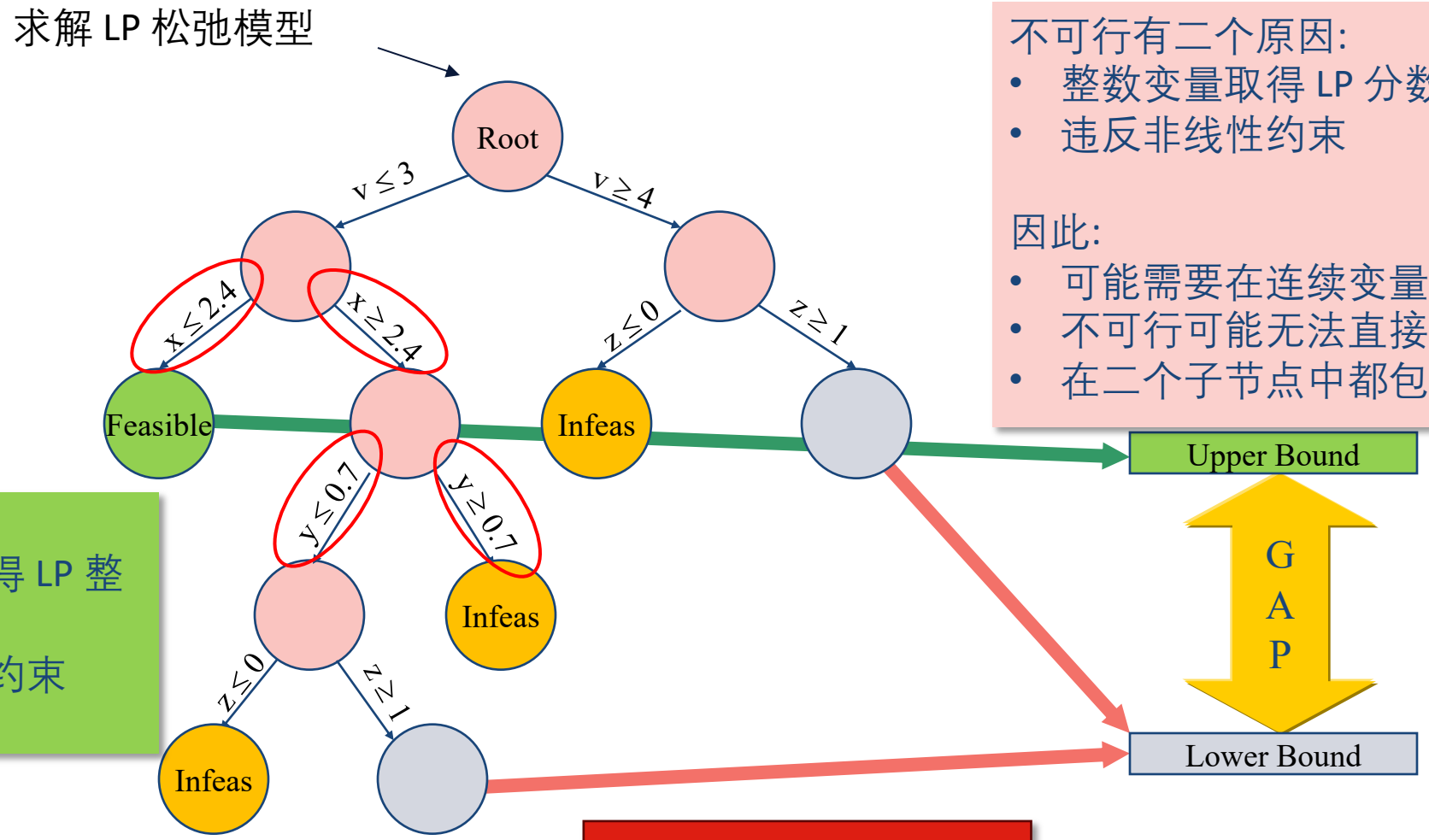


GUROBI OPTIMIZATION

“过大” 范围

- 如果 x 取值范围过大，很难从松弛中获得有用信息
- 对 x 分支可以快速收紧松弛问题！
- 缩小初始上下界可以加快求解速度

MINLP 的空间分支定界



不可行有二个原因:

- 整数变量取得 LP 分数值
- 违反非线性约束

因此:

- 可能需要在连续变量上分支
- 不可行可能无法直接解决
- 在二个子节点中都包含分支分割点

可行解:

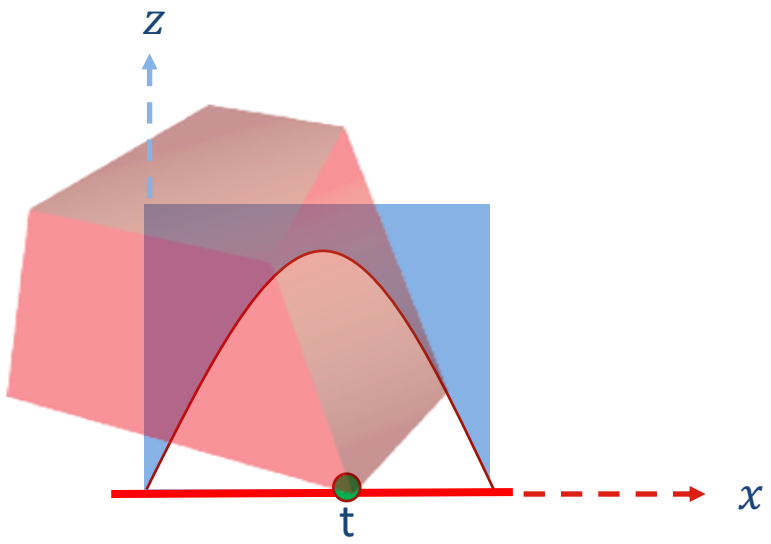
- 整数变量取得 LP 整数数值
- 满足非线性约束

如何松弛?
如何分支?

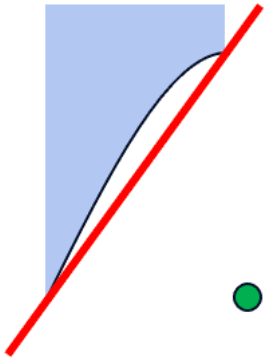
Branching 分支

- 求解了凸松弛之后，如何在违反的非凸约束上分支？

非凸
$-z - x^2 \leq 0$

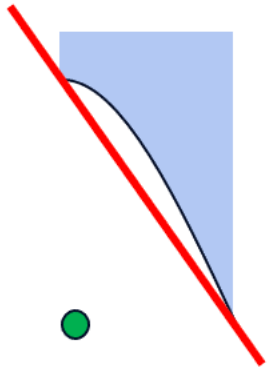


$$\begin{array}{lllllll} \min & c^T x & + & d^T z & & & \\ \text{s.t.} & Ax & + & Dz & \leq & b & \\ & -x_i x_j & + & z_{ij} & = & 0 & \text{for all } (i,j) \in S \\ & l & \leq & x & \leq & u & \\ & & & x_j & \in & \mathbb{Z} & \text{for all } j \in I \end{array}$$



分支
$x \leq t$ or $x \geq t$

本地更新松弛边界
和相关McCormick包络



复合非线性函数

- Gurobi 11.0 可以处理单变量非线性约束 $f(x) = y$
  - 三角函数，幂函数，对数函数，指数函数等
  - 这些做为组合单元可以构造更复杂的非线性函数
- 举例: 假设我们希望构造

$$f(\mathrm{x}) = \underbrace{\sqrt{\underbrace{1 + \mathrm{x}^2}_{u = 1 + \mathrm{x}^2}}}_{v = \sqrt{u}} + \ln\Big(\underbrace{\mathrm{x} + \underbrace{\sqrt{1 + \mathrm{x}^2}}_{v}}_{w = \mathrm{x} + \mathrm{v}}\Big) \leq 2,\ \mathrm{x} \geq 0 \qquad z = \ln w$$

- 我们引入辅助变量 $u, v, w, z \geq 0$ 和如下约束:
  - $\mathrm{u} = 1 + \mathrm{x}^2,\ u = v^2,\ w = \mathrm{x} + v,\ z = \ln w$
  - 那么 $f(\mathrm{x}) \leq 2$ 可以表示为 $v + z \leq 2$

MINLP 在 Gurobi 12 的开发计划

- 增加 API 函数，可以直接表达复合非线性函数
  - 利用复合非线性函数进行可行性验证
  - 利用复合非线性函数进行内点法 LP 求解器求解
  - 利用复合非线性的知识更好进行预优化和外逼近算法
- 改进全局精确 MINLP 算法性能
  - 预优化缩减
  - 切平面
  - 改进启发式算法更好地适应非线性约束
  - 更好分支和分割点选择
- 内点法局部 NLP 求解器
  - 提供给用户我们内部局部NLP 求解器接口
    - 用户可以提供一个局部最优解
  - 改进局部 NLP 求解器的性能和稳定性
- 改进数值问题

GUROBI OPTIMIZATION

API 接口改进: Java & Gurobipy

Java API

- Java 包的名字修改为 `com.gurobi.gurobi` 不再是 `gurobi`
  - 遵从 Java 标准命名格式
- Java 包支持由 Maven Central 分发
  - 广受欢迎的 Java 包仓库
  - 类似 Python 中的 PyPI
  - 帮助 Java 用户更好地管理编译和部署流程

Python gurobipy 安装流程变化

GUROBI OPTIMIZATION

- 自带类型提示辅助包
  - 无需再安装 `gurobipy-stubs`

```
The conflict is caused by:
    The user requested gurobipy==11.0.0b1
    gurobipy-stubs 2.0.0 depends on gurobipy==10.*
    The user requested gurobipy==11.0.0b1
    gurobipy-stubs 1.0.1.post0 depends on gurobipy==9.5.*
```

- 不再提供 `setup.py install`
  - pip 提供了离线安装方法
  - pip 提供了Hash 验证

- conda 和 pip 兼容更好
  - 不再出现重复安装
  - 对于Gurobi 开源包来说，conda 安装更清爽

```
# Name                    Version                   Build  Channel
blas                      1.0                         mkl
bottleneck                1.3.5           py311hb9e55a9_0
bzip2                     1.0.8                h1de35cc_0
ca-certificates           2023.08.22           hecd8cb5_0
gurobi                    10.0.3                  py311_0    gurobi
gurobipy                  10.0.3                  pypi_0     pypi
gurobipy-pandas           1.0.0                   pypi_0     pypi
intel-openmp              2023.1.0        ha357a0b_43547
```

Python gurobipy 矩阵 API 变化

GUROBI OPTIMIZATION

- Callback 回调函数可以接受矩阵变量和矩阵约束

```
x_sol = model.cbGetSolution(x)
model.cbLazy(A @ x <= b)
```

- 基于Numpy 规则的合并操作 (hstack, vstack, concatenate)

```
X = model.addMVar((n, m))
Y = model.addMVar((n, k))
XY = gp.concatenate((X, Y), axis=1)  # (n, m+k) MVar
```

- 矩阵友好的指示约束 (vectorized, broadcastable)

```
z = model.addVar(vtype=GRB.BINARY)
x = model.addMVar(n)
model.addGenConstrIndicator(z, True, A @ x <= b)  # MGenConstr ...
```

Python gurobipy 其他明显变化

- Callback 可以是任何 callable 对象
  - 允许Callback 是可调用的类
  - 避免 `model._attribute` 变通做法
  - 用户可以参考更新后的 tsp.py 和 callback.py 范例
- 速度提升
  - `addConstr(A @ x == b)` 对于稀疏数据有~2x 提速
  - 基于分项组合的建模方式有 ~10-20% 提速 (感谢 Cython 开发人员这方面的贡献)
- 参考《详细发布指南》获得全部更新内容

```
52 class TSPCallback:
53     """Callback class implementing lazy constraint
54     callbacks, solutions are checked for subtours
55     constraints are added if needed."""
56
57     def __init__(self, nodes, x):
58         self.nodes = nodes
59         self.x = x
60
61     def __call__(self, model, where):
62         """Callback entry point: call lazy constra
63         solutions are found. Stop the optimization
64         user code."""
65         if where == GRB.Callback.MIPSOL:
66             try:
67                 self.eliminate_subtours(model)
68             except Exception:
69                 logging.exception("Exception occur
70                 model.terminate()
71
```

GUROBI
OPTIMIZATION

Q&A