Gurobi 入门和进阶网络课程
课程三：Gurobi 高级操作和使用方法

刃之砺信息科技（上海）有限公司

课程大纲

(1) Callback使用方法

用户有时候在求解过程中需要实现一些功能,例如获取一些信息、终止优化、添加约束条件(割平面)、嵌入自己的算法等。

(2) 常用的线性化方法

Maxmin/Minmax 目标函数

Maxmax/Minmin 目标函数

带fixed cost目标函数

分式目标函数

逻辑或

乘积式

Partial integer variable

Callback使用方法

- Callback函数

Callback为用户在求解模型时提供了高级控制功能。它允许用户在Gurobi**求解过程中**获取信息、终止优化、加入额外约束条件(割平面)、加入自己开发的算法等。

定义CallBack函数： **def 函数名(model, where): ……**

调用CallBack函数： **m.optimize(callback函数名)**

CallBack函数使用时需要注意两个重要的参数：

where: 回调函数触发点

what: 获取何种信息, what能够获取什么取决于where

Callback使用方法

- where 参数取值

where	数值	优化器状态
POLLING	0	轮询回调
PRESOLVE	1	预处理
SIMPLEX	2	单纯形
MIP	3	当前Mip
MIPSOL	4	发现新的Mip解
MIPNODE	5	当前探索节点
MESSAGE	6	打印出Log信息
BARRIER	7	当前内点法
MULTIOBJ	8	当前多目标

Callback使用方法

- what参数取值

what 值取决于 where 的取值(**使用时一定要正确对应两者关系**), 例如 where = MIP 时, what 取值如下：

what	类型	描述
MIP_OBJBST	double	当前最优目标值
MIP_OBJBND	double	当前最优界
MIP_NODCNT	double	当前已探索的节点数
MIP_SOLCNT	int	当前发现可行解的数量
MIP_CUTCNT	int	当前割平面使用次数
MIP_NODLFT	double	当前未搜索的节点数
MIP_ITRCNT	double	当前单纯形迭代步数

Callback使用方法

- what参数取值

当 where = MIPSOL 时, what的取值如下：

what	类型	描述
MIPSOL_SOL	double*	当前解的具体取值
MIPSOL_OBJ	double	新解的目标值
MIPSOL_OBJBST	double	当前最优目标值
MIPSOL_OBJBND	double	当前最优界
MIPSOL_NODCNT	double	当前以搜索的节点数
MIPSOL_SOLCNT	int	当前发现可行解的数量

如果Where = MIPSOL, what 取 MIP_OBJBST 就会报错。

Callback使用方法

- Callback函数

查询一些信息，例如目标值，节点数等。使用时注意 what 与 where 的匹配。

cbGet(what)

- what 获取何种信息，what 能够获取什么取决于where

例如，查询当前单纯形的目标函数值。

```
def mycallback(model, where):
    if where == GRB.Callback.SIMPLEX:
        print(model.cbGet(GRB.Callback.SPX_OBJVAL))
model.optimize(mycallback)
```

Callback使用方法

- Callback函数

查询变量在当前节点的松弛解。

注意 where == GRB.Callback.MIPNODE 且 GRB.Callback.MIPNODE_STATUS == GRB.OPTIMAL 才起作用。

cbGetNodeRel (vars) **函数返回：变量在当前节点的松弛解**

  - vars 需要查询的变量

```
def mycallback(model, where):
    if where == GRB.Callback.MIPNODE and model.cbGet(GRB.Callback.MIPNODE_STATUS)== GRB.OPTIMAL :
        print model.cbGetNodeRel(model._vars)
model._vars = model.getVars()
model.optimize(mycallback)
```

callback函数可以通过“_变量名” 获得外部变量的值。

Callback使用方法

- Callback函数

查询可行解变量的取值。

注意 where == GRB.Callback.MIPSOL 或者 GRB.Callback.MULTIOBJ。

cbGetSolution (vars) **函数返回：变量在新可行解中的取值**

  - vars 需要查询的变量

```
def mycallback(model, where):
    if where == GRB.Callback.MIPSOL:
        print (model.cbGetSolution(model._vars))
model._vars = model.getVars()
model.optimize(mycallback)
```

Callback使用方法

- Callback函数

  在节点添加割平面。

  注意 where == GRB.Callback.MIPNODE 参数 **PreCrush=1** (关掉Gurobi预处理对模型约束的转化)。

  cbCut (lhs, sense, rhs)

  - lhs 左端项
  - sense 符号
  - rhs 右端项

Callback使用方法

- Callback函数

例如，通过节点的松弛解信息构造割平面。

```
def mycallback(model, where):
    if where == GRB.Callback.MIPNODE:
        status = model.cbGet(GRB.Callback.MIPNODE_STATUS)
        if status == GRB.OPTIMAL:
            rel = model.cbGetNodeRel([model._vars[0], model._vars[1]])
                if rel[0] + rel[1] > 1.1:
                    model.cbCut(model._vars[0] + model._vars[1] <= 1)
model._vars = model.getVars()
model.Params.PreCrush = 1
model.optimize(mycallback)
```

Callback使用方法

- Callback函数

在节点添加 Lazy cut。(与一般cut区别在于只有在被违反的时候才起作用)

注意 Where == GRB.Callback.MIPNODE or GRB.Callback.MIPSOL

使用时必须设定参数LazyConstraints = 1

cbLazy(lhs, sense, rhs)

- lhs 左端项
- sense 符号
- rhs 右端项

Callback使用方法

- Callback函数

例如，通过可行解的信息构造Lazy cut 。

```
def mycallback(model, where):
    if where == GRB.Callback.MIPSOL:
        sol = model.cbGetSolution([model._vars[0], model._vars[1]])
        if sol[0] + sol[1] > 1.1:
            model.cbLazy(model._vars[0] + model._vars[1] <= 1)
model._vars = model.getVars()
model.Params.lazyConstraints = 1
model.optimize(mycallback)
```

Callback使用方法

- Callback函数

向当前节点导入一个解(完整或部分都可以)。

注意 where == GRB.Callback.MIPNODE

cbSetSolution (vars, solution)

- vars 变量
- solution 变量的值

cbUseSolution ( )

计算导入解的目标值。

Callback使用方法

- Callback函数

```
def mycallback(model, where):
    if where == GRB.Callback.MIPNODE:
        model.cbSetSolution(vars, newsolution)
model.optimize(mycallback)
```

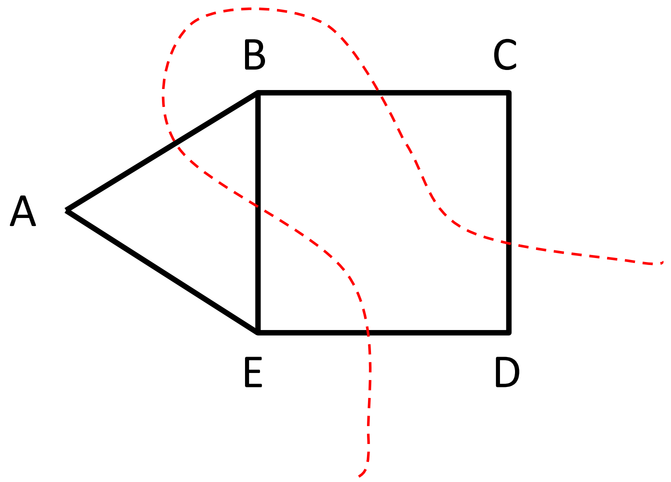
对复杂的问题，可以开发启发式算法找到高质量的解，然后导入解让Gurobi在其基础上继续求解。

Callback使用方法

- Callback案例

最大割问题(Maximum cut) + RINS heuristic

假设图中线段都被赋上了权重，希望找到一种方案将顶点分成两个子集(记为m,n)，使得属于不同子集点的连线权重和最大。例如，图中的方案m = {A, E,C}， n = {B, D}， 权重总和 = AB+BE+BC+CD+ED。



Callback使用方法

- Callback案例

最大割问题(Maximum cut)模型

参数： C_{ij} 线段 i,j 权重，N 顶点数量

变量： $x_i = 1,\ if\ i \in m\ \ or\ \ x_i = -1,\ if\ i \in n$

$$max\ \ \frac{1}{4}\sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij}(1 - x_i x_j)$$

$$x_i \in \{-1,1\}$$

$$\boldsymbol{x = 2y - 1}, \qquad \mathbf{y \in \{0, 1\}}$$

Callback使用方法

- Callback案例

RINS heuristic 核心思想

随着整数规划模型的求解进程，**节点松弛模型的解与最优解之间的差距可能会越来越小，体现在松弛解的部分变量值与最优解对应变量值相等或差距很小。**因此利用松弛模型的信息可能会更快发现高质量的可行解。

RINS heuristic 基于上面的想法，在求解过程中抓取节点松弛解（可能是部分整数，部分小数），固定模型中对应的变量的取值构造一个子模型（规模往往远小于原模型），然后求解子模型。如果发现了更好的可行解，把解传递给优化器让其在它的基础上继续求解原模型。

Callback使用方法

- Callback案例代码

$$\boldsymbol{x} = \mathbf{2}\boldsymbol{y} - \mathbf{1}\,, \boldsymbol{y} \in \{\mathbf{0}, \mathbf{1}\} \ \Rightarrow$$

$$max \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij}(y_i + y_j - 2y_i y_j)$$

```
N = 20
#随机产生线段权重
random.seed(1)
Cmatrix = {(i,j):random.randint(0,100) for i in range(N) for j in range(N)}
m = Model('MaximumCut')
#添加变量
y = m.addVars(N, vtype=GRB.BINARY, name='y')
#构造目标函数
obj = QuadExpr()
for i in range(N):
    for j in range(N):
        obj = obj+Cmatrix[i,j]*(y[i]+y[j]-2*y[i]*y[j])
m.setObjective(0.5*obj, -1)
#设置求解时间
m.Params.TimeLimit = 600
#外部变量
m._y = y
m._N = N
#求解
m.optimize(RINScallback)
#获得目标值和变量值
print("Obj = ",m.ObjVal)
for i in range(N):
    print(y[i].VarName,' = ',y[i].x)
```

Callback使用方法

- Callback案例代码

```
def RINScallback(model, where):
    if where == GRB.Callback.MIPNODE:
        # MIP node callback
        if model.cbGet(GRB.Callback.MIPNODE_NODCNT) % 100 == 0 and\
          model.cbGet(GRB.Callback.MIPNODE_STATUS) == GRB.OPTIMAL:
            submodel = model.copy()
            suby = submodel.getVars()
            #获得节点松弛解
            yrelaxation = model.cbGetNodeRel(model._y)
            #固定变量取值
            for i in range(model._N):
                if abs(yrelaxation[i])<0.01:
                    suby[i].ub = 0
                elif abs(yrelaxation[i]-1)<0.01:
                    suby[i].lb = 1
            submodel.setParam(GRB.Param.TimeLimit, 30)
            submodel.setParam(GRB.Param.OutputFlag, 0)
            submodel.optimize()
            if submodel.objval > model.cbGet(GRB.Callback.MIPNODE_OBJBST):
                #将解传递给原模型
                for i in range(model._N):
                    if abs(suby[i].x)<0.001:
                        model.cbSetSolution(model._y[i], 0.0)
                    elif abs(suby[i].x-1)<0.001:
                        model.cbSetSolution(model._y[i], 1.0)
```

常用线性化方法

- 广义约束—Max/Min

$$\boldsymbol{z = max(x, y, 3)}$$

线性化方法：

取值	0	1
u_1	$\boldsymbol{x \geq z - M}$ 恒成立	$x \geq z$
u_2	$\boldsymbol{y \geq z - M}$ 恒成立	$y \geq z$
u_3	$\boldsymbol{3 \geq z - M}$ 恒成立	$3 \geq z$

$$x \leq z, y \leq z, 3 \leq z$$
$$x \geq z - M(1 - u_1)$$
$$y \geq z - M(1 - u_2)$$
$$3 \geq z - M(1 - u_3)$$
$$u_1 + u_2 + u_3 \geq 1$$
$$u_1, u_2, u_3 \in \{0,1\}$$

$x \geq z, y \geq z, 3 \geq z$

至少有一个约束成立

按照 Gurobi 广义约束的写法，可以直接写为： `m.addConstr(z==max_(x, y, 3))`

常用线性化方法

- 广义约束—Max/Min

$$\boldsymbol{z = min(x, y, 3)}$$

线性化方法：

$$x \geq z, y \geq z, 3 \geq z$$

$$x \leq z - M(1 - u_1)$$

$$y \leq z - M(1 - u_2)$$

$$3 \leq z - M(1 - u_3)$$

$$u_1 + u_2 + u_3 \geq 1$$

$$u_1, u_2, u_3 \in \{0,1\}$$

按照 Gurobi 广义约束的方法，可以直接写为： `m.addConstr(z==min_(x, y, 3))`

常用线性化方法

- 目标函数中存在绝对值

$$\boldsymbol{min} \sum_i \boldsymbol{c_i |x_i|}$$

$$Ax = b$$

$$x_i \ \ free, \boldsymbol{c_i \geq 0}, \forall i$$

线性化方法1：$y_i = |x_i| , y_i \geq x_i , y_i \geq -x_i$

$$min \sum_i c_i y_i$$

$$Ax = b$$

$$y_i \geq x_i , y_i \geq -x_i$$

$$x_i \ \ free, c_i > 0$$

常用线性化方法

- 目标函数中存在绝对值

线性化方法2：$\forall\, x,\ \exists\, u, v \geq 0$, 使得 $x = u - v, |x| = u + v$, 其中 $u = \frac{|x|+x}{2}, v = \frac{|x|-x}{2}$

$$min \sum_i c_i(u_i + v_i)$$

$$A(u - v) = b$$

$$u, v\ \geq 0$$

按照 Gurobi 广义约束的方法，可以直接写为： m.addConstr(y==abs_(x))

常用线性化方法

- Maxmin/Minmax目标函数

$$\max(\min_{k\in K} \sum_i c_{ki}x_i)$$

线性化方法：

$$\max z$$

$$z \le \sum_i c_{ki}x_i, \forall k \in K$$

$$\min(\max_{k\in K} \sum_i c_{ki}x_i)$$

线性化方法：

$$\min z$$

$$z \ge \sum_i c_{ki}x_i, \forall k \in K$$

常用线性化方法

- Maxmin/Minmax目标函数

$$\boldsymbol{max(min\begin{matrix} x+2y+10 \\ 3x+y+1 \end{matrix})}$$

线性化方法：

$$max\ z$$

$$z \le x + 2y + 10$$

$$z \le 3x + y + 1$$

常用线性化方法

- Maxmax/Minmin目标函数

$$\boldsymbol{max}(\boldsymbol{max}_{k\in K}\sum_{i}\boldsymbol{c}_{ki}\boldsymbol{x}_i)$$

线性化方法：

$$max\ z$$

$$\sum_i c_{ki}x_i \geq z - M(1-y_k), \forall k \in K$$

$$\sum_k y_k \geq 1$$

$$y_k \in \{0,1\}$$

常用线性化方法

- Maxmax/Minmin目标函数

$$\boldsymbol{max}\left(\boldsymbol{max}\begin{matrix}\boldsymbol{x+2y+10}\\\boldsymbol{3x+y+1}\end{matrix}\right)$$

线性化方法：

$$max\ z$$

$$x+2y+10\geq z-M(1-u)$$

$$3x+y+1\geq z-M(1-v)$$

$$u+v\geq 1$$

$$u,v\in\{0,1\}$$

可以去掉: $x+2y+10\leq z$

$3x+y+1\leq z$

常用线性化方法

- Maxmax/Minmin目标函数

$$\boldsymbol{min(min_{k\in K}\sum_{i} c_{ki}x_i)}$$

线性化方法：

$$min\ z$$

$$\sum_i c_{ki}x_i \leq z + M(1 - y_k),\forall k \in K$$

$$\sum_k y_k \geq 1$$

$$y_k \in \{0,1\}$$

常用线性化方法

- 带fixed cost目标函数

$$\min f(x) = \begin{cases} 0, & x = 0 \\ cx + k, & x > 0, k > 0 \end{cases}$$

线性化方法：

$$\min cx + ky$$
$$x \le My$$
$$y \in \{0,1\}$$

常用线性化方法

- 分式目标函数

$$\min \sum_i (c_i x_i + \alpha) \,/\, \sum_i (d_i x_i + \beta)$$

$$\sum_i a_{ij} x_i \le b_j,\ \ \forall j \in J$$

$$\sum_i d_i x_i + \beta > 0,\ x_i \ge 0, \forall j \in J$$

线性化方法：令 $y = \frac{1}{\sum_i (d_i x_i + \beta)} > 0$

$$\min \sum_i (c_i x_i y + \alpha y)$$

$$\sum_i a_{ij} x_i \le b_j,\ \ \forall j \in J$$

$$\sum_i d_i x_i y + \beta y = 1,\ \ \forall j \in J$$

$$y > 0, x_i \ge 0, \forall j \in J$$

$z_i = x_i y$

$$\min \sum_i (c_i z_i + \alpha y)$$

$$\sum_i a_{ij} z_i \le b_j y,\ \ \forall j \in J$$

$$\sum_i d_i z_i + \beta y = 1,\ \ \forall j \in J$$

$$y > 0,\ z_i \ge 0, \forall j \in J$$

常用线性化方法

- 分式目标函数

$$\boldsymbol{min}\frac{2\boldsymbol{x}+\boldsymbol{y}+\mathbf{1}}{\boldsymbol{x}+\mathbf{3}\boldsymbol{y}}$$

$$\mathbf{5}\boldsymbol{x}+\boldsymbol{y}\leq\mathbf{6}$$
$$\boldsymbol{x}+\mathbf{3}\boldsymbol{y}>\mathbf{0},\boldsymbol{x},\boldsymbol{y}\geq\mathbf{0}$$

线性化方法：令 $z=\frac{1}{x+3y}>0$

$$min(2x+y)\,z+z$$
$$5x+y\leq 6$$
$$z(x+3y)=1$$
$$x,y\geq 0,z>0$$

$$u=xz,v=yz$$
$$5xz+yz\leq 6z$$

$$min\,2u+v+z$$
$$5u+v\leq 6z$$
$$u+3v=1$$
$$z>0,u,v\geq 0$$

常用线性化方法

- 逻辑或

$$\sum_j \boldsymbol{a_{1j}\, x \le b_1} \text{ 或 } \sum_j \boldsymbol{a_{2j}\, x \le b_2} \quad \text{(两个约束至少一个成立)}$$

线性化方法：

$$\sum_j a_{ij}\, x \le b_i + M(1-y_i), i = 1,2$$

$$\sum_i y_i \ge 1$$

$$y_i \in \{0,1\}, i = 1,2$$

常用线性化方法

- 逻辑或

$$x + 2y + 10 \leq 15 \text{ 或 } 3x + y + 1 \leq 5$$

线性化方法：

$$x + 2y + 10 \leq 15 + M(1 - u)$$

$$3x + y + 1 \leq 5 + M(1 - v)$$

$$u + v \geq 1$$

$$u, v \in \{0,1\}$$

常用线性化方法

- 逻辑或

$$\sum_j \boldsymbol{a_{1j}}\, \boldsymbol{x} \le \boldsymbol{b}_1 \text{ 或 } \sum_j \boldsymbol{a_{2j}}\, \boldsymbol{x} = \boldsymbol{b}_2 \quad \text{(两个约束至少一个成立)}$$

线性化方法：

$$\sum_j a_{ij}\, x \le b_i + M(1 - y_i), i = 1,2$$

$$\sum_j a_{2j}\, x \ge b_2 - M(1 - y_2)$$

$$\sum_i y_i \ge 1$$

$$y_i \in \{0,1\}, i = 1,2$$

常用线性化方法

- 逻辑或

$$x+2y+10\leq 15 \text{ 或 } 3x+y+1=5$$

线性化方法：

$$x+2y+10\leq 15+M(1-u)$$

$$3x+y+1\leq 5+M(1-v)$$

$$3x+y+1\geq 5-M(1-v)$$

$$u+v\geq 1$$

$$u,v\in\{0,1\}$$

常用线性化方法

- 逻辑或

$$\boldsymbol{\sum_j a_{1j}\, x = b_1} \text{ 或 } \boldsymbol{\sum_j a_{2j}\, x = b_2} \quad \text{(两个约束至少一个成立)}$$

线性化方法：

$$\sum_j a_{ij}\, x \le b_i + M(1 - y_i), i = 1,2$$

$$\sum_j a_{ij}\, x \ge b_i - M(1 - y_i), i = 1,2$$

$$\sum_i y_i \ge 1$$

$$y_i \in \{0,1\}, i = 1,2$$

常用线性化方法

- 逻辑或

$$x + 2y + 10 = 15 \text{ 或 } 3x + y + 1 = 5$$

线性化方法：

$$x + 2y + 10 \le 15 + M(1 - u)$$

$$x + 2y + 10 \ge 15 - M(1 - u)$$

$$3x + y + 1 \le 5 + M(1 - v)$$

$$3x + y + 1 \ge 5 - M(1 - v)$$

$$u + v \ge 1$$

$$u, v \in \{0,1\}$$

常用线性化方法

- 乘积式

$$\boldsymbol{x_1 x_2}, \text{其中 } \boldsymbol{x_1, x_2} \in \{\mathbf{0}, \mathbf{1}\}$$

线性化方法：$\mathrm{y} = \boldsymbol{x_1 x_2}$

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	y
0	0	0
0	1	0
1	0	0
1	1	1

$$y \le x_1$$

$$y \le x_2$$

$$y \ge x_1 + x_2 - 1$$

$$y \in \{0,1\}$$

常用线性化方法

- 乘积式

$$\boldsymbol{x_1 x_2}, \text{其中 } \boldsymbol{x_1} \in \{\mathbf{0}, \mathbf{1}\}, \boldsymbol{x_2} \in [\mathbf{0}, \boldsymbol{u}]$$

线性化方法： $\mathrm{y} = \boldsymbol{x_1 x_2}$

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	y
0	[0,u]	0
1	[0,u]	$\boldsymbol{x_2}$

$$y \le u x_1$$

$$y \le x_2$$

$$y \ge x_2 - u(1 - x_1)$$

$$y \in [0, u]$$

常用线性化方法

- 乘积式

$$\boldsymbol{x_1 x_2}, \text{其中 } \boldsymbol{x_1} \in \{\mathbf{0}, \mathbf{1}\}, \boldsymbol{x_2} \in [\boldsymbol{l}, \boldsymbol{u}]$$

线性化方法： $y = \boldsymbol{x_1 x_2}$

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	y
0	[l,u]	0
1	[l,u]	$\boldsymbol{x_2}$

$$y \le x_2$$

$$y \ge x_2 - u(1 - x_1)$$

$$lx_1 \le y \le ux_1$$

常用线性化方法

- Partial integer variable

$$\boldsymbol{z \in [0, a]\ integer \quad or \quad z \in [a, b]\ continuous}$$

线性化方法： $x \in [0, a]\ integer$

$y \in [a, b]\ continuous$

$u \in \{0,1\}$

$$z \le x + (1-u)M$$
$$z \ge x - (1-u)M$$
$$z \le y + uM$$
$$z \ge y - uM$$

谢谢各位对 Gurobi 中文网络课程的支持。

我们会不断推出专题培训,敬请关注

www.gurobi.cn