



Gurobi 入门和进阶网络课程

(课程一： Gurobi 基本操作)

刃之砺信息科技有限公司（上海）有限公司

www.gurobi.cn



大纲

- (1) 安装 Gurobi 软件和 Anaconda IDE 软件，建立可视化 Python 建模环境。
- (2) 结合案例，介绍采用 Python 语言建立优化模型并求解。



为什么 Gurobi ?

Gurobi 是目前数学规划（线性和凸二次规划）优化器性能领袖、性价比领袖。

全球用户超过1600家。国内应用涵盖航空运输、电力、制造、传媒管理、生物医药、通讯、金融等。

性价比更优，技术支持响应更快。

学术许可免费，申请便捷（www.gurobi.cn）

可以求解大规模线性问题，二次型目标问题和混合整数线性和二次型问题

支持多目标优化

支持包括SUM, MAX, MIN, AND, OR等广义约束和逻辑约束

支持并行计算和分布式计算

提供了方便轻巧的接口，支持 C++, Java, Python, .Net, Matlab 和R，内存消耗少

支持多种平台，包括 Windows, Linux, Mac OS X



哪里寻找可靠的 Gurobi 资料

(1) 安装目录 examples, docs 目录下有完整的范例和使用手册、参考手册

(2) 官网 www.gurobi.com

(2.1) 在线手册: <http://www.gurobi.com/documentation/>

(2.2) 视频: <http://www.gurobi.com/resources/seminars-and-videos/seminars-videos>

(2.3) 在线课程: <http://www.gurobi.com/academia/for-online-courses>

(3) 中文网站 www.gurobi.cn

建议不要网上搜索片段化、来源不明、过时的文档



为什么 Python?

很好的编程语言

容易学习，可读性好

跨平台

适合初学者和专业编程人员

简洁、紧凑的语法

开源社区：海量可用的模块库、代码和资料

很棒的数学建模工具

很多受欢迎的数学计算工具库

Jupyter Notebook: 交互式图形化环境

有很多语言表述方式和数据结构，和数学表达式很接近

可扩展性



为什么 Python?

Sep 2018	Sep 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.436%	+4.75%
2	2		C	15.447%	+8.06%
3	5	▲	Python	7.653%	+4.67%
4	3	▼	C++	7.394%	+1.83%
5	8	▲	Visual Basic .NET	5.308%	+3.33%
6	4	▼	C#	3.295%	-1.48%
7	6	▼	PHP	2.775%	+0.57%
8	7	▼	JavaScript	2.131%	+0.11%
9	-	▲	SQL	2.062%	+2.06%
10	18	▲	Objective-C	1.509%	+0.00%
11	12	▲	Delphi/Object Pascal	1.292%	-0.49%
12	10	▼	Ruby	1.291%	-0.64%
13	16	▲	MATLAB	1.276%	-0.35%
14	15	▲	Assembly language	1.232%	-0.41%
15	13	▼	Swift	1.223%	-0.54%
16	17	▲	Go	1.081%	-0.49%
17	9	⚡	Perl	1.073%	-0.88%
18	11	⚡	R	1.016%	-0.80%
19	19		PL/SQL	0.850%	-0.63%
20	14	⚡	Visual Basic	0.682%	-1.07%



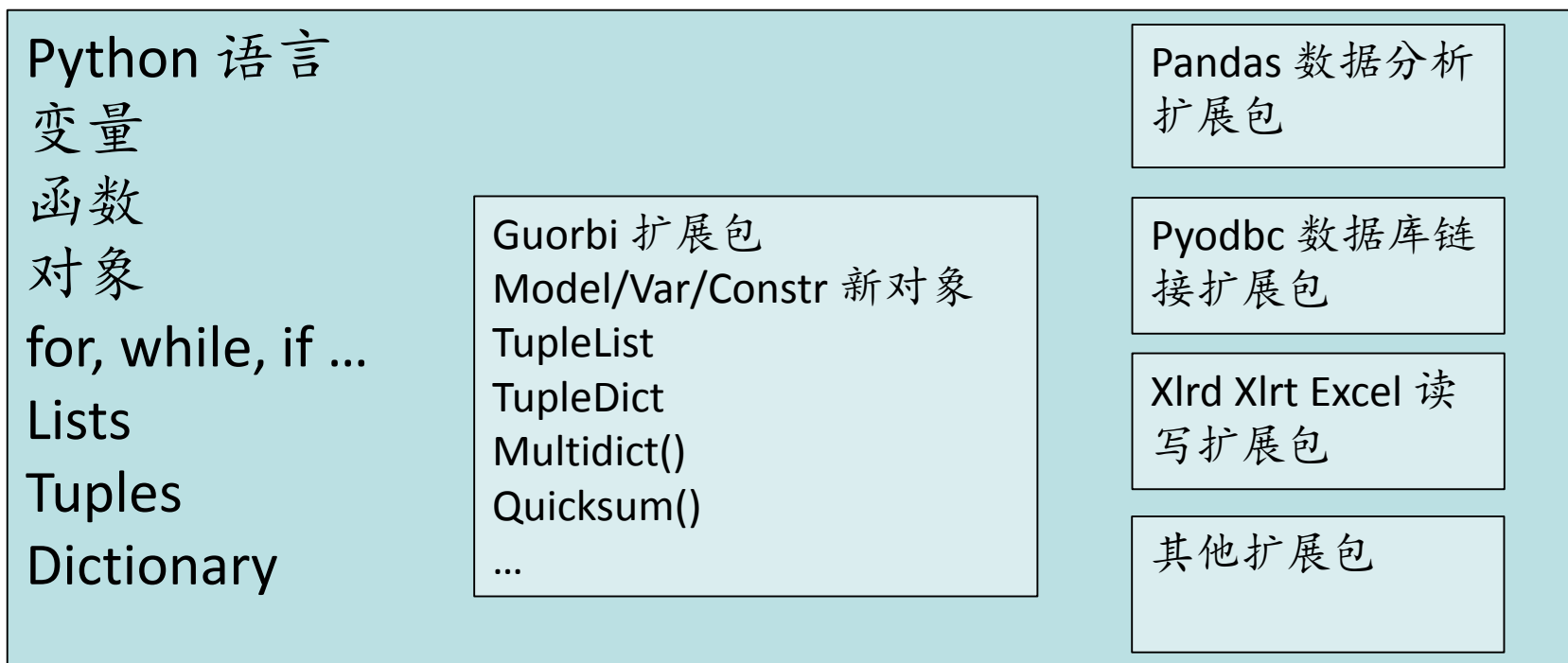
哪里寻找可靠的 Python 资料

- (1) 官网 <https://wiki.python.org/moin/BeginnersGuide>
- (2) 培训视频 <https://www.python.org/doc/av/>

使用 Gurobi 并不需要具备很深的 Python 知识。



Gurobi Python 关系



Gurobi 扩展语法和数据结构的目的:

- (1) 建立Gurobi 特有的数学模型对象 Model/Var/Constr
- (2) 提升数学建模中常见操作（下标搜索和选择、求和、系数相乘等）操作的效率要求



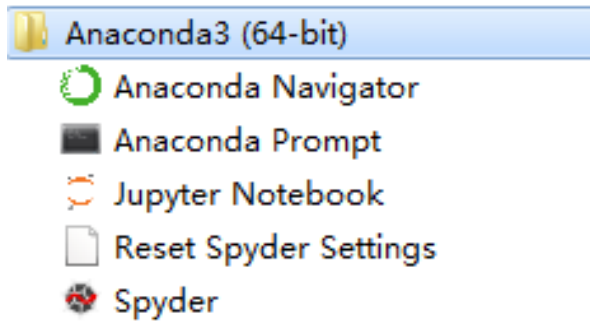
为什么 Anaconda?

最受欢迎的支持 Python 和 R 的数据科学开发、测试、训练平台

支持多种操作系统

一站式管理平台：管理扩展包，管理项目环境

带有多个常用工具：Spyder, Jupyter Notebook 等





哪里寻找可靠的 Anaconda 资料

- (1) 官网 <https://enterprise-docs.anaconda.com/en/latest/>
- (2) 培训视频 <https://www.anaconda.com/videos/>

使用 Gurobi 并不需要具备很深的 Anaconda 知识。



选择 Python 2.x 还是 Python 3.x

Python 2.x 将于 2020年1月1日停止支持。 <https://pythonclock.org/>

Gurobi 支持 Python 2.7, 3.5, 3.6, 3.7

建议选择 Anaconda 3.x 版本



使用 Gurobi + Python 的几种方式

安装包 (1) : Gurobi 安装包, 下载地址 www.gurobi.com 或者 QQ 群 251135672 的群文件

安装包 (2) : Python 安装包, 下载地址 www.python.org

安装包 (3) : Anaconda 安装包, 下载地址 www.anaconda.com

方式	安装方式	使用方式	利	弊
1	(1) Gurobi 里调 Python	gurobi.bat xxxx.py gurobi 交互界面 (Python)	简单	非 IDE 开发, 界面 不友好
2	(1) + (2) Python 里调 Gurobi	双击 <installdir>/bin/pysetup (Windows) 将 gurobipy 模块安装到 python 安装目录下	自由选择 Python IDE 环境, 例如 Canopy, Eric, iep, PyCharm, and PyDev.	需要更深知识
3	(3) 或者 (3) + (1)	conda install gurobi 将 gurobipy 安装 到 Anaconda 环境中	简单管理, 丰富 工具	学习更多工具

推荐方式 3



使用 Anaconda + Gurobi 的安装方式

- (1) Gurobi 单独安装包可以安装，也可以不安装。推荐安装：有帮助文档和案例；如果开发其他语言则需要
- (2) 根据操作系统，选择下载 Anaconda 3.x 适合版本 (www.anaconda.com)，并安装
- (3) 进入到 Anaconda Prompt（类似于 Dos 命令）（有些机器需要以管理员权限打开）
- (4) 利用 conda 扩展包管理工具，进行扩展包的安装，更新，删除等



使用 Anaconda + Gurobi 的安装方式

如果联网：

添加 Gurobi 安装路径

conda config --add channels <http://conda.anaconda.org/gurobi>

安装 Gurobi 扩展包

conda install gurobi

更新Gurobi 扩展包

conda update gurobi

删除 Gurobi 扩展包

conda remove gurobi

查看已经安装的扩展包

conda list



使用 Anaconda + Gurobi 的安装方式

如果不联网：

下载 Gurobi Conda 离线扩展包（注意区分 Python 版本和操作系统）

<https://anaconda.org/Gurobi/gurobi/files>

通过 cd 命令进入到扩展包下载目录

安装 Gurobi 扩展包

`conda install gurobi-8.0.1-py36h0b08b80_0.tar.bz2`（更换成合适的文件名字）

删除 Gurobi 扩展包

`conda remove gurobi`

查看已经安装的扩展包

`conda list`



产生许可文件

如果没有安装 Gurobi 独立安装包，那么需要在 Anaconda Prompt 中运行；如果安装了 Gurobi 独立安装包，也可以在普通 Dos 命令下（CMD）运行，

```
grbgetkey XXXXXXXX-XXXXXX-XXXX-XXXX-XXXXXXXXXXXXXXXXXX
```

需要在线激活。如果 IP 地址不识别为学校，则需要通过 www.gurobi.cn 的步骤申请免 IP 验证的学术许可。

产生后的许可文件 gurobi.lic 放置在 c:\gurobi（Windows）或者 opt/gurobi（Linux）目录下，如果没有这些目录，可以创建一个。以后不管安装多少版本，许可文件都可以放在这个目录下，集中管理。



推荐 Jupyter Notebook

网页应用程序，创建实时交互内容，可以包含实时数据、图表、实时代码、排版公式、文档等丰富内容。

源于 Ipython 项目（2014）

包含在 Anaconda 中

广泛用户基础

在 Anaconda Prompt 下，cd 进入到工作目录，然后运行
jupyter notebook



介绍 Spyder

Python 可视化集成开发工具

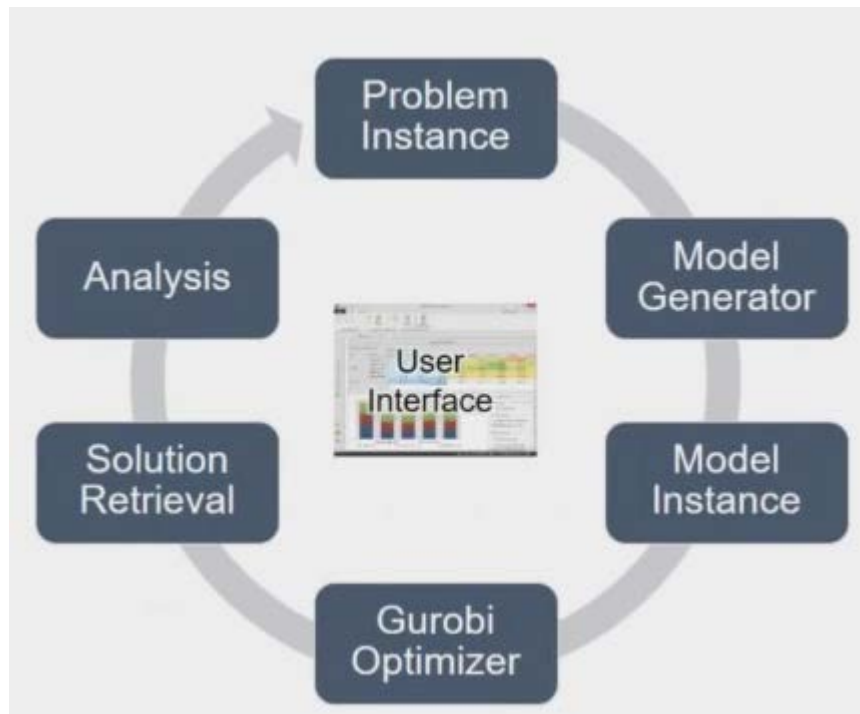
界面友好的代码编辑器

调试工具和交互工具

帮助文档



Gurobi 建模过程



Problem Instance: 待优化问题

Model Generator: 将数据组合成模型，
产生计算机模型对象

Model Instance: 存在于内存的一个完整
数学模型

Gurobi Optimizer: Gurobi 优化求解

Solution Retrieval: 根据需要读取优化结
果

Analysis: 对结果进行分析
循环往复，直到获得满意结果



Gurobi 建模基本概念

Parameter（参数）控制优化器的行为，需要在优化启动前设置。

Attributes（属性）控制模型（包括模型、变量、约束、目标等对象）的特性。

例如：

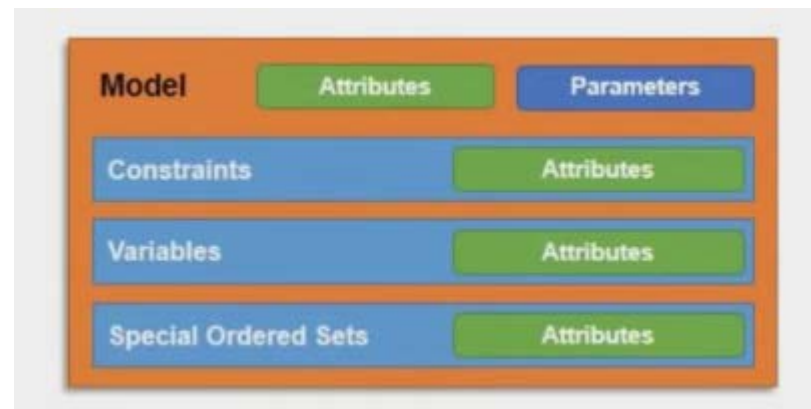
模型 ModelSense

变量 LB/UB

约束 RHS

Environment 是包含模型和全局参数的一个容器，也是许可控制的节点。

Python 中的环境会自动建立。





Python 语法

网上已经有大量培训资料和视频 (www.python.org)。Gurobi 建模对于 Python 语言本身要求不高。

(1) 变量

```
x=5  
y="abc"
```

(2) 对象、属性和方法

```
m=Model("myModel")  
m.ModelSense=
```

(3) for, while 循环

(4) if .. elif



Python 语法

Python List（列表） 用 `[]` 表示，适合做下标、索引、变量、约束等各种对象的集合

一个排序的集合

- `cities = ['A', 'B', 'C', 'D']`

可以修改、添加、删除，排序

- Ex: Add, delete, sort elements

虽然 Python 有 `set` 类型，但 List 更有效率



Python 语法

Python Tuple（元组），用()表示

固定、不可修改的组合

- link = ('A', 'B')

适合做多维下标



Python 语法

Python Dictionaries 字典

Python 中的字典可以通过任意**键值**对数据进行映射。任何无法修改的 Python 对象都可以当作键值使用：整数、浮点数、字符串、和元组。

适合用来表示带下标的数值，例如变量

举个例子，以下这些语句创建了一个字典对象 `x`，并将数值 1 与键值 ('Pens', 'Denver', 'New York') 进行关联：

```
gurobi> x = {} # 创建一个空白的字典对象
gurobi> x[('Pens', 'Denver', 'New York')] = 1
gurobi> print x[('Pens', 'Denver', 'New York')] # 1
```

在 Python 中，通过元组访问字典时可以省略圆括号，因此以下语句同样也是同样有效的：

```
gurobi> x = {}
gurobi> x['Pens', 'Denver', 'New York'] = 2
gurobi> print x['Pens', 'Denver', 'New York'] # 2
```




Python 语法

Python List/Tuple/Dictionary 的不足

在建模过程中，经常要对带下标数据做挑选，不同下标的数据进行组合，这样面临着二个处理方法

- (1) 全部循环。多维下标意味着多重循环+ if 条件
这样的处理方法没有效率
- (2) 采用特殊的Gurobi 扩展对象 TupleList 和 TupleDict

Results of tuplelist speed test

N	# elements (K)	Dense count	Tuplelist count
100	1050	0.36 sec	0.0006 sec
200	2100	3.00 sec	0.0013 sec
400	4200	24.17 sec	0.0025 sec
800	8400	195.26 sec	0.0049 sec



Python 语法

Gurobi tuplelist -- Python list 的扩展对象

```
Cities= [('A','B'), ('A','C'), ('B','C'),('B','D'),('C','D')]  
Routes = tuplelist(Cities)
```

使用 tuplelist() 不能忘记 from gurobipy import *
tuplelist 增加了快速筛选 select 功能

```
print(Routes.select('A','*'))
```

```
Result=[]  
for i,j in Cities:  
    if i=='A':  
        Result.append((i,j))  
print(Result)
```



Python 语法

Gurobi tupledict – Python Dictionary 扩展

键值为 tuple（元组），可以使用 select, sum, prod 函数

用于变量和约束（后面案例中体现）



Python 语法

Multidict() 创建 tuplelist 和 tupledict 的便捷方法

```
cities, supply, demand = multidict({  
'A': [100, 20],  
'B': [150, 50],  
'C': [20, 300],  
'D': [10, 200]})
```

返回 cities (tuplelist)、supply (tupledic) 和 demand (tupledict)



Python 语法

创建 list （列表解析 List Comprehension）

```
a=[]
```

```
a.append('A')
```

```
b = [i**2 for i in range(6)] # [0,1,4,9,16,25]
```

```
c =[(i,j) for j in range(4) for i in range(j)] #[(0,1), (0,2), (1,2), (0,3), (1,3), (2,3)]
```

```
d = [i for i in range(10) if i not in b] # [2,3,5,6,7,8]
```

```
Pairs=[]
```

```
for j in range(4):
```

```
    for i in range(j):
```

```
        Pairs.append((i,j))
```



Python 语法

Generator (生成器)

```
SumSquares = sum(i**2 for i in range(6)) # 55
```

Gurobi 中采用 **quicksum**, 效率更高

```
obj = quicksum(cost[i,j]*x[i,j] for i,j in arcs)
```



Python 语法

tupledict (Gurobi 变量一般都是 tupledict 类型) 有 sum 函数

```
from gurobipy import *  
m=Model() 两个维度，一个是0, 1, 2, 3，另一个是0, 1, 2, 3  
x=m.addVars(3,4, vtype=GRB.BINARY, name="x")  
m.addConstrs((x.sum(i,'*')<=1 for i in range(3)), name="con")  
m.update()  
m.write("test.lp")
```

产生如下约束

$$\begin{aligned}x[0,0] + x[0,1] + x[0,2] + x[0,3] &\leq 1 \\x[1,0] + x[1,1] + x[1,2] + x[1,3] &\leq 1 \\x[2,0] + x[2,1] + x[2,2] + x[2,3] &\leq 1\end{aligned}$$



Python 语法

tupledict (Gurobi 变量一般都是 tupledict 类型) 有 prod 函数, 用于变量和系数相乘后累加。

以下二个表达式等效。(x 和 cost 要有相同的键值)

```
obj = quicksum(cost[i,j] * x[i,j] for i,j in arcs)
obj = x.prod(cost)
```




Python 语法

建模建议，尽量采用稀疏方式

采用 tuplelists 筛选和指定合适的下标组合关系

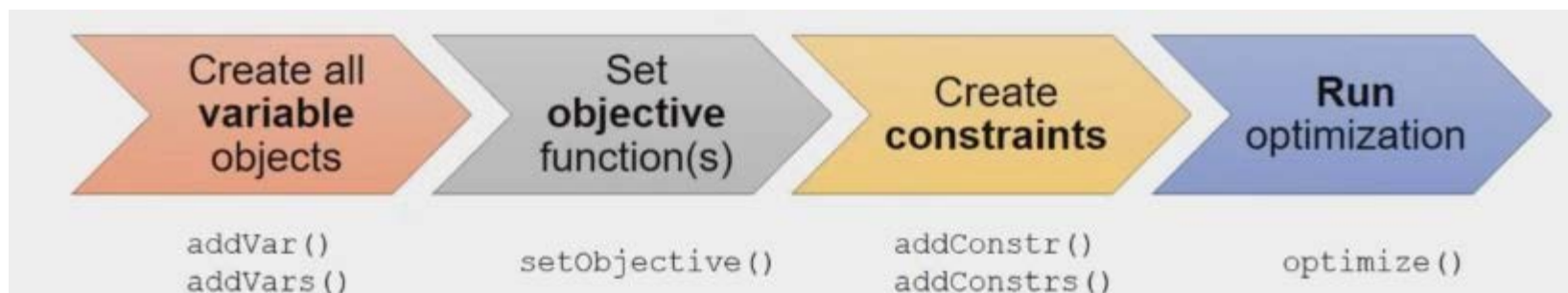
基于这些组合关系建立变量和数据字典

利用 `tuplelist.select()` 以及 `tupledict.select()`, `tupledict.sum()`, `tupledict.prod()` 来

对下标进行组合处理



Gurobi 建模过程





m=Model()

$$\begin{aligned} \max \quad & \sum_{i \in I, j \in J} c_{ij} \cdot x_{ij} \\ \text{s.t.} \quad & \sum_{i \in I} x_{ij} \leq 1 \quad \forall j \in J \\ & \sum_{j \in J} x_{ij} \leq 1 \quad \forall i \in I \\ & x_{ij} \text{ binary} \quad \forall i \in I, j \in J \end{aligned}$$



m.addVar()

m.addVars()

$$\begin{array}{ll} \max & \sum_{i \in I, j \in J} c_{ij} \cdot x_{ij} \\ \text{s.t.} & \sum_{i \in I} x_{ij} \leq 1 \quad \forall j \in J \\ & \sum_{j \in J} x_{ij} \leq 1 \quad \forall i \in I \\ & x_{ij} \text{ binary} \quad \forall i \in I, j \in J \end{array}$$

Decision Variables



m.setObjective()

$$\begin{array}{llll} \max & \sum_{i \in I, j \in J} c_{ij} \cdot x_{ij} & & \text{Objective} \\ \text{s.t.} & \sum_{i \in I} x_{ij} & \leq & 1 \quad \forall j \in J \\ & \sum_{j \in J} x_{ij} & \leq & 1 \quad \forall i \in I \\ & x_{ij} & \text{binary} & \forall i \in I, j \in J \end{array}$$



m.addConstr()
m.addConstrs()

$$\max \sum_{i \in I, j \in J} c_{ij} \cdot x_{ij}$$

Constraints

s.t.

$$\sum_{i \in I} x_{ij}$$

\leq

1

$$\forall j \in J$$

$$\sum_{j \in J} x_{ij}$$

\leq

1

$$\forall i \in I$$

$$x_{ij}$$

binary

$$\forall i \in I, j \in J$$



m.addVar() / m.addVars()
m.addConstr() / m.addConstrs()
m.setObjective()

$$\begin{aligned} \max \quad & \sum_{i \in I, j \in J} c_{ij} \cdot x_{ij} && \text{Data coefficients} \\ \text{s.t.} \quad & \sum_{i \in I} x_{ij} \leq 1 && \forall j \in J \\ & \sum_{j \in J} x_{ij} \leq 1 && \forall i \in I \\ & x_{ij} \text{ binary} && \forall i \in I, j \in J \end{aligned}$$



Lists

TupleList

$$\begin{array}{llll} \max & \sum_{i \in I, j \in J} c_{ij} \cdot x_{ij} & & \text{Index sets} \\ \text{s.t.} & \sum_{i \in I} x_{ij} & \leq & 1 \quad \forall j \in J \\ & \sum_{j \in J} x_{ij} & \leq & 1 \quad \forall i \in I \\ & x_{ij} & \text{binary} & \forall i \in I, j \in J \end{array}$$



TupleList.select()

$$\begin{aligned} \max \quad & \sum_{i \in I, j \in J} c_{ij} \cdot x_{ij} \\ \text{s.t.} \quad & \sum_{i \in I} x_{ij} \leq 1 \quad \forall j \in J \\ & \sum_{j \in J} x_{ij} \leq 1 \quad \forall i \in I \\ & x_{ij} \text{ binary} \quad \forall i \in I, j \in J \end{aligned}$$



+ - *

tupledict.prod()

$$\max \sum_{i \in I, j \in J} c_{ij} \cdot x_{ij}$$

Arithmetic operators

$$\text{s.t.} \quad \sum_{i \in I} x_{ij} \leq 1 \quad \forall j \in J$$

$$\sum_{j \in J} x_{ij} \leq 1 \quad \forall i \in I$$

$$x_{ij} \text{ binary} \quad \forall i \in I, j \in J$$



\leq, \geq

GRB.LESS_EQUAL, GRB.EQUAL, or
GRB.GREATER_EQUAL

$$\begin{array}{llll} \max & \sum_{i \in I, j \in J} c_{ij} \cdot x_{ij} & & \text{Constraint operators} \\ \text{s.t.} & \sum_{i \in I} x_{ij} & \leq & 1 \quad \forall j \in J \\ & \sum_{j \in J} x_{ij} & \leq & 1 \quad \forall i \in I \\ & x_{ij} & \text{binary} & \forall i \in I, j \in J \end{array}$$



List Comprehension 列表解析 中的 for 语句

$$\begin{array}{llll} \max & \sum_{i \in I, j \in J} c_{ij} \cdot x_{ij} & & \text{For-all operators} \\ \text{s.t.} & \sum_{i \in I} x_{ij} \leq 1 & \forall j \in J & \\ & \sum_{j \in J} x_{ij} \leq 1 & \forall i \in I & \\ & x_{ij} & \text{binary} & \forall i \in I, j \in J \end{array}$$



quicksum

$$\begin{array}{ll} \max & \sum_{i \in I, j \in J} c_{ij} \cdot x_{ij} \\ \text{s.t.} & \sum_{i \in I} x_{ij} \leq 1 \quad \forall j \in J \\ & \sum_{j \in J} x_{ij} \leq 1 \quad \forall i \in I \\ & x_{ij} \text{ binary} \quad \forall i \in I, j \in J \end{array}$$

Aggregate sum operators



Gurobi 建模举例 1 (<examples/python/mip1.py>)

$$\begin{array}{ll}\max & x + y + 2z \\ \text{s.t.} & x + 2y + 3z \leq 4 \\ & x + y \geq 1 \\ & x, y, z \in \{0, 1\}\end{array}$$



Gurobi 建模举例 2 营养配方模型 (examples/python/diet.py)

人体需要四种营养: category : calories, protein, fat, sodium

食物来源: foods= hamburger, chicken, hot dog, fries, macaroni, pizza, salad, milk, ice cream

营养吸收每天有上限和下限

单位重量食物价格不同

单位重量食物所含营养成分不同

求达到足够营养花费的代价最小



从外部文件读入数据

数据量比较大的时候，可以直接从 Excel, ODBC 数据库等读入，然后用 multidict, list.append, list[n]=xxx, dict[k1,ke..]=value 等方式赋值。

参考 example/python 目录下的 diet 系列模型



下次课程中将会介绍

- (1) Gurobi 常用的重要参数和属性
- (2) 自动参数调优
- (3) 广义约束的表达方式和使用
- (4) 多目标优化的表达方式和使用
- (5) 多个解集合的实现方式和使用

感谢参加。请抽出时间仔细阅读软件自带的范例和手册