

整数规划

分支定界算法 & 割平面算法

杉数科技·求解器团队·皇甫琦

2022年4月



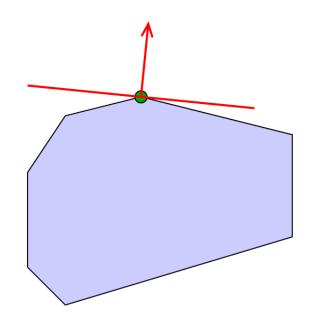




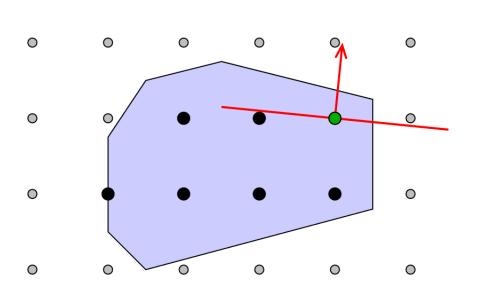
- 简介
- ・分支定界算法
- 割平面算法
- 建模实例
- 拓展话题

■ 混合整数规划简介

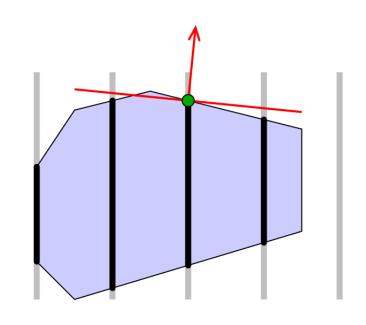




线性规划 Linear programming (LP)



整数规划 Integer programming (IP)



混合整数规划 Mixed-Integer programming (MIP)

■ 混合整数规划简介 – 数学形式



max
$$f = c^T x$$

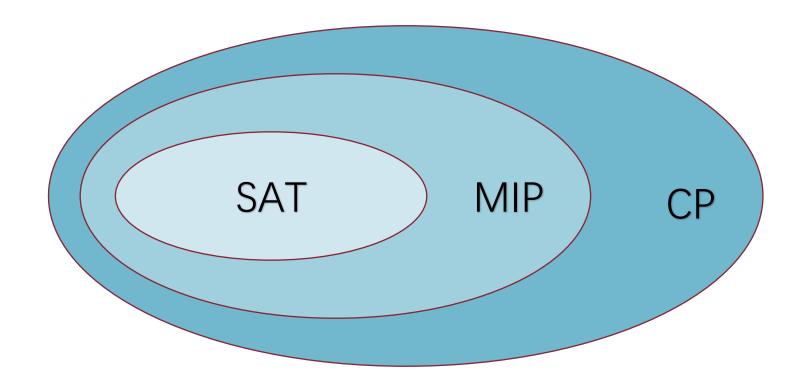
s.t. $Ax = b$
 $x \ge 0$
 $x_{i \in I} \in \mathbb{Z}$

其中 $x, c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $I \subseteq \{1, 2, ..., n\}$

- *I* = {1,2,...,*n*} 整数规划
- *I* ⊂ {1,2,...,*n*} 混合整数规划

■ 混合整数规划简介 – SAT vs MIP vs CP





■ 混合整数规划简介 – 如何求解



特殊情况

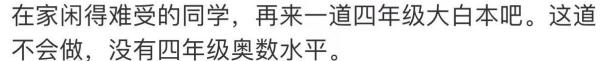
- 针对问题的专有算法
- 线性规划的解就是整数解

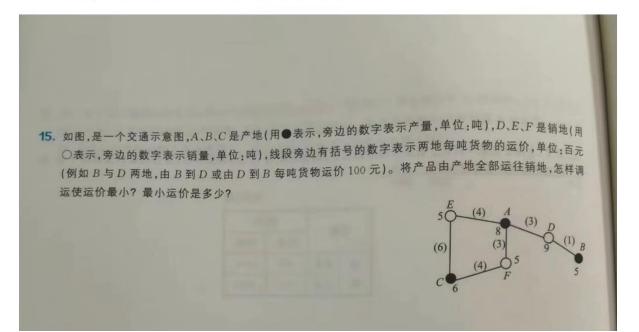












网络线性规划 (简化版)

Minimize
3 AD + 3 AF + 4 AE + 1 BD + 6 CE + 4 CF
Subject To
AE + AD + AF = 8 # 产地A输出
BD = 5 # 产地B输出
CE + CF = 6 # 产地C输出
AD + BD = 9 # 销地D输入
AE + CE = 5 # 销地E输入
AF + CF = 5 # 销地F输入
End

最小运价

59 (百元)

运量

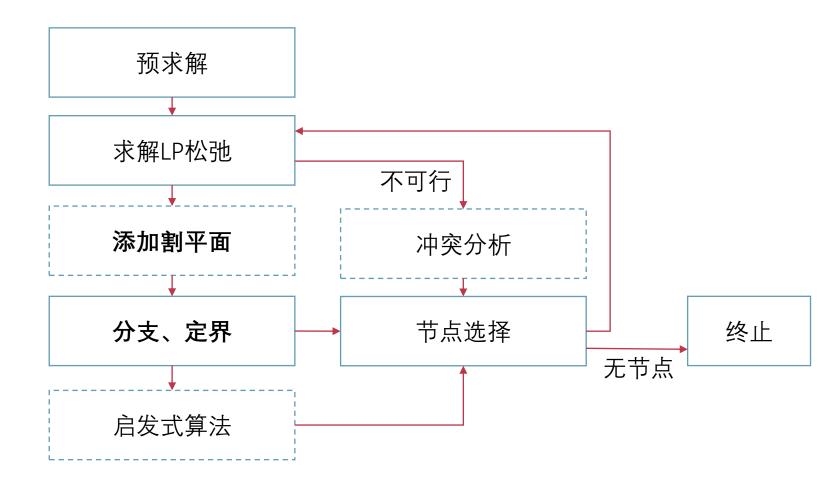
$$AD = 4$$
, $AF = 0$, $AE = 4$
 $BD = 5$, $CE = 1$, $CF = 5$

■ 混合整数规划简介 – 如何求解



通用算法

- 预求解
- 求解LP松弛
 - 单纯形法
 - 内点法
- 分支定界
- 割平面
- 启发式算法
- 冲突分析



参考文献:

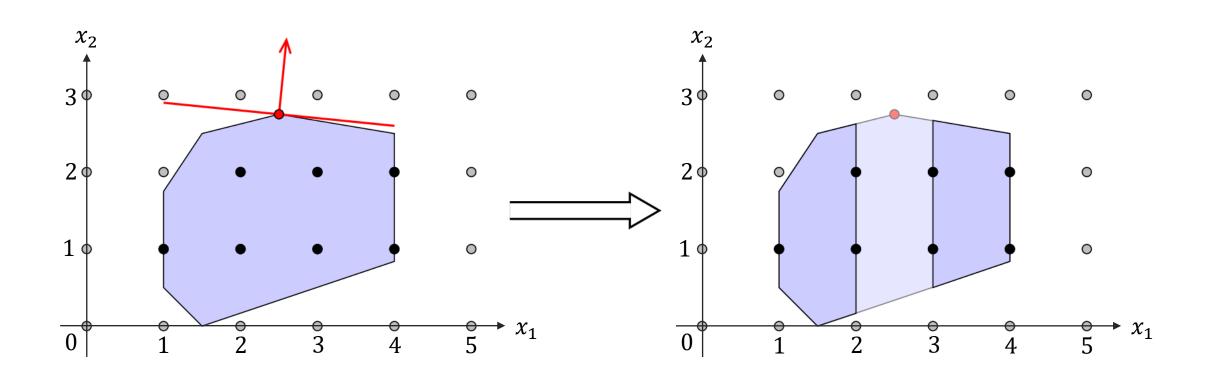
Achterberg T. Constraint Integer Programming[D]., 2007. Conforti M, Cornuéjols G, Zambelli G. Integer programming[M]. Berlin: Springer, 2014.

■分支定界算法



基于线性规划的分支定界算法

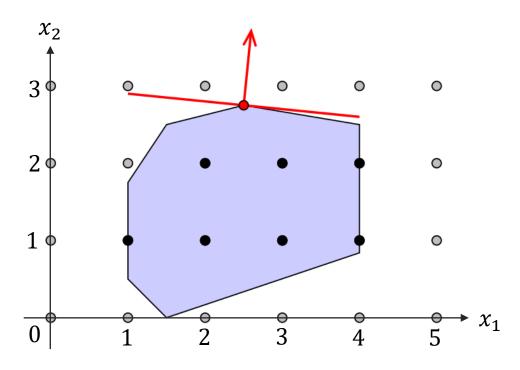
选定一个取值不为整数的变量,将问题分为两个子问题



■分支定界算法

杉数科技 Cardinal Operations

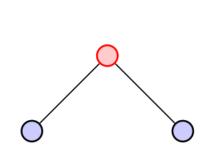
求解LP松弛 算得 x_1, x_2 均不是整数

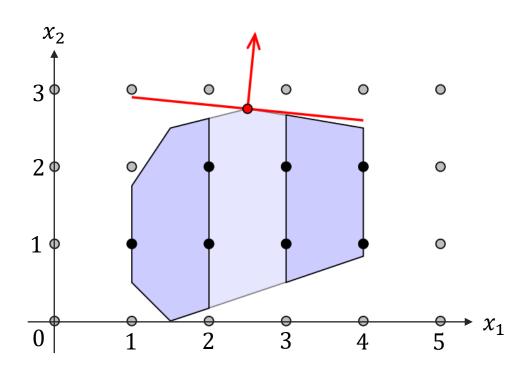




分支

选定 x_1 进行分支,分为 $x_1 \le 2, x_1 \ge 3$ 两个部分



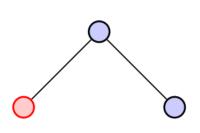


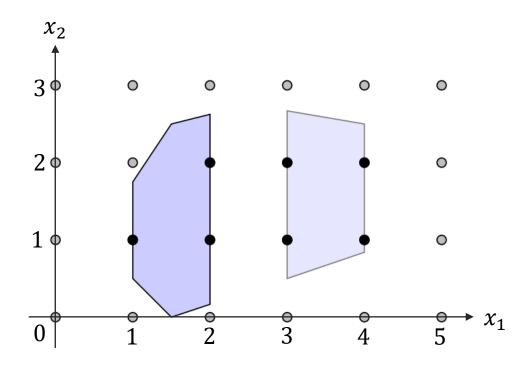




选定子节点

选定 $x_1 \leq 2$

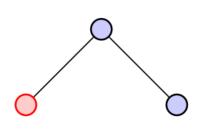


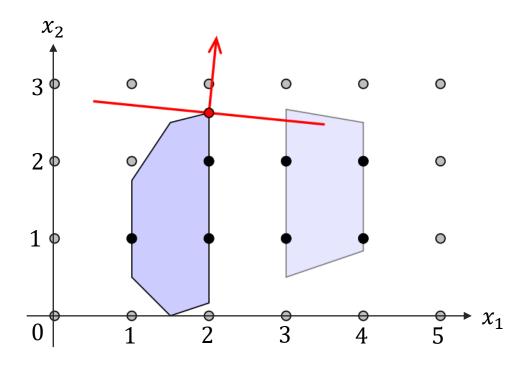


■分支定界算法



计算LP松弛 算得 x_2 不是整数



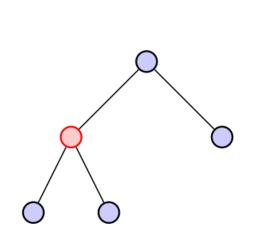


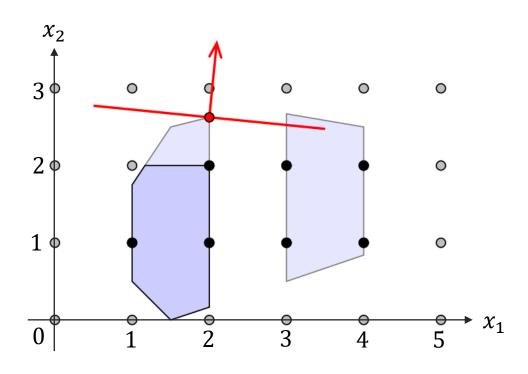
▶分支定界算法



分支

选定 x_2 进行分支,分为 $x_2 \le 2, x_2 \ge 3$ 两个部分



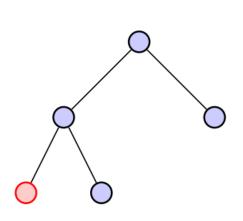


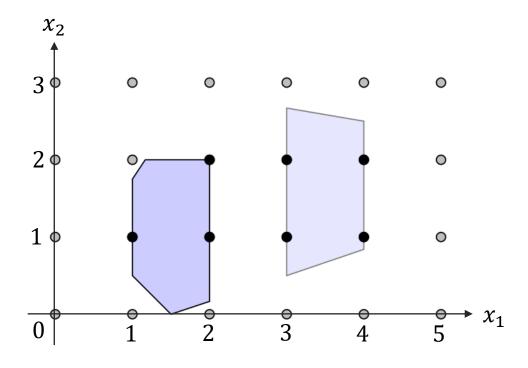
■分支定界算法



选定子节点

选定 $x_1 \le 2, x_2 \le 2$

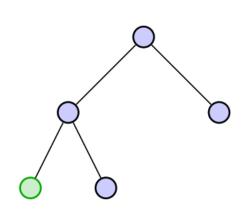


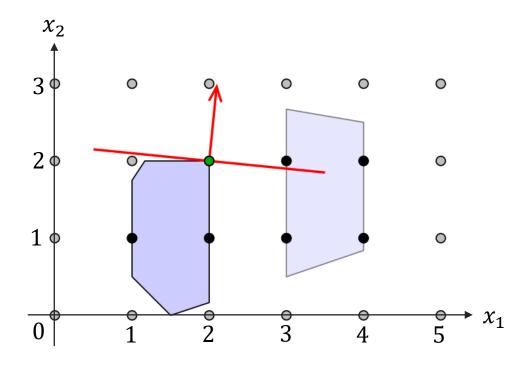




求解LP松弛

求得整数解
$$x_1 = 2, x_2 = 2$$



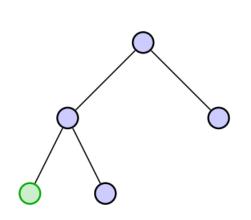


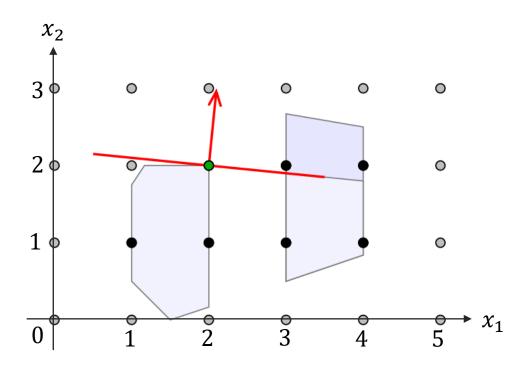
▶分支定界算法



定界

用当前整数解切除所有子问题的可行域空间

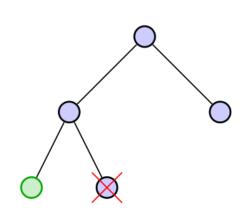


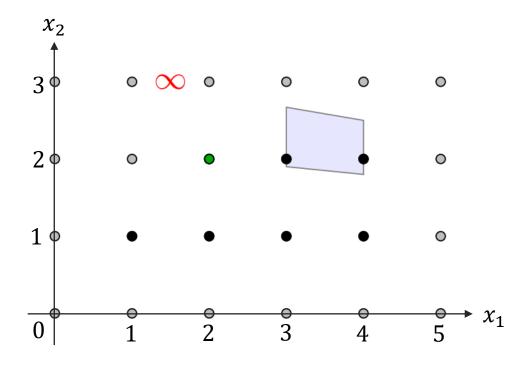


杉数科技 Cardinal Operations

选定子节点

选定 $x_1 \le 2, x_2 \ge 3$, LP不可行



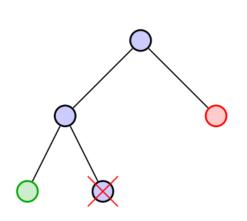


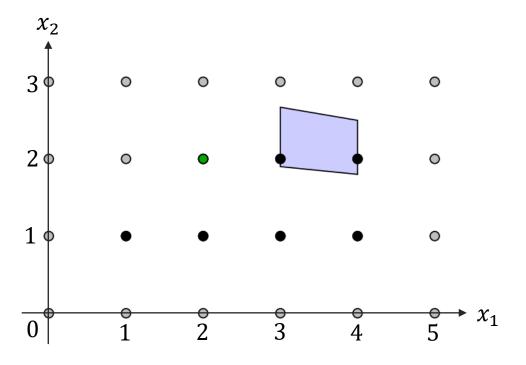


杉数科技 Cardinal Operations

选定子节点

选定 $x_1 \ge 3$

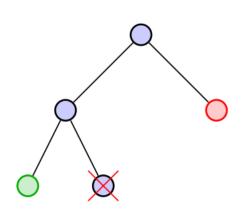


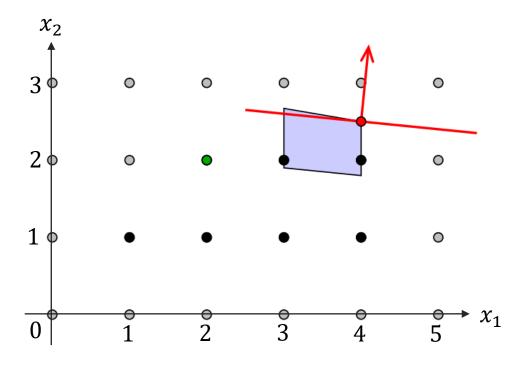




杉数科技 Cardinal Operations

计算LP松弛 算得 x_2 不是整数



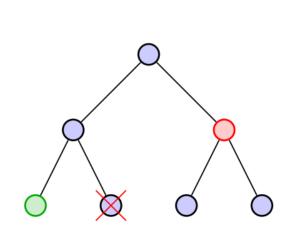


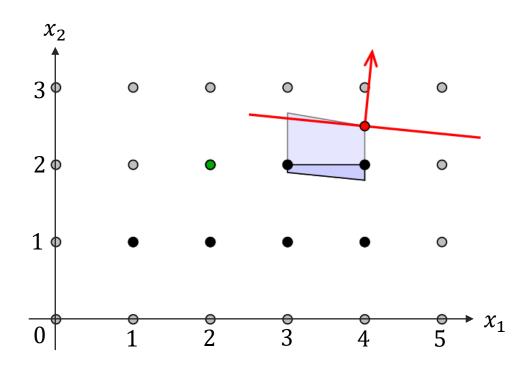
▶分支定界算法



分支

选定 x_2 进行分支,分为 $x_2 \le 2, x_2 \ge 3$ 两个部分



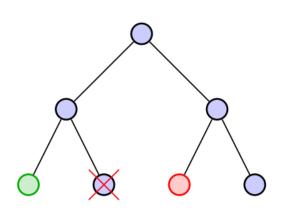


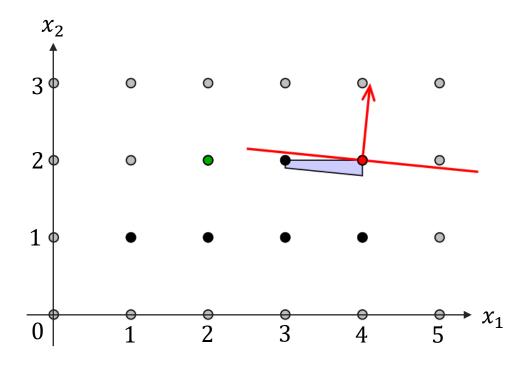
■分支定界算法



求解LP松弛

求得整数解
$$x_1 = 4, x_2 = 2$$

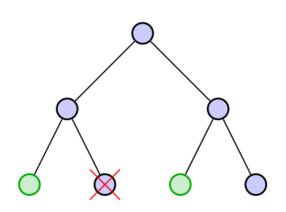


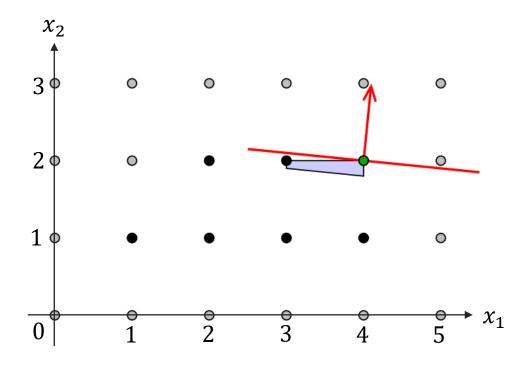


▶分支定界算法



更新最佳整数解

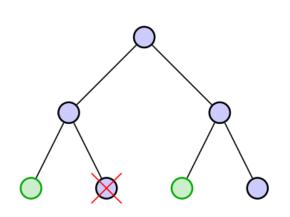


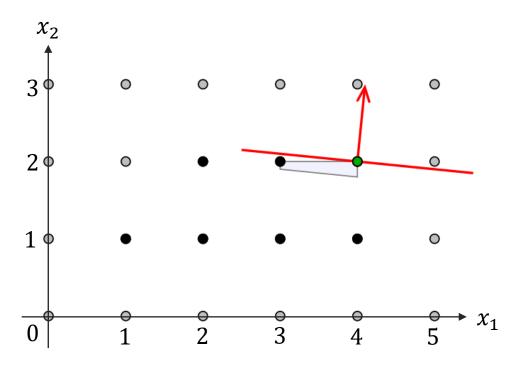




定界

用当前整数解切除所有子问题的可行域空间

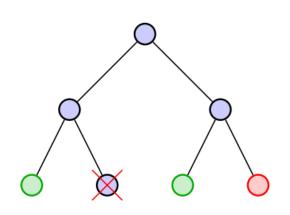


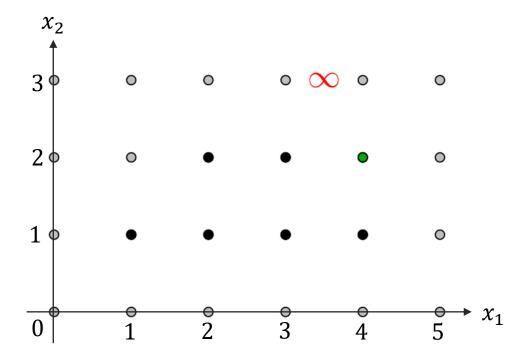


杉数科技 Cardinal Operations

选定子节点

选定 $x_1 \ge 3$, $x_2 \ge 3$, LP不可行

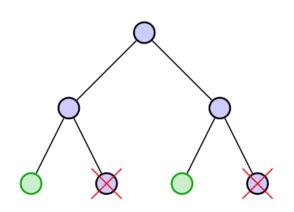


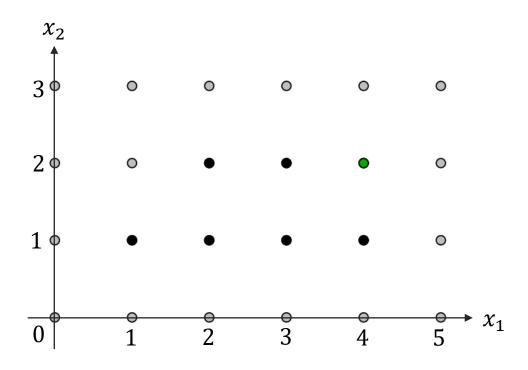




最终结果

完成搜索,求得最优整数解 $x_1 = 4, x_2 = 2$





■ 分支定界算法 – 算法流程



1. 初始化

节点集合 $N = \{ \text{初始LP松弛} \}, f^* = -\infty, x^* = \emptyset$

2. 中止测试

当 $N = \emptyset$ 时, $x^* \neq \emptyset$ 就是最优解 (或 $x^* = \emptyset$ 整数规划不可行)

3. 选择节点

从N 选择并取出一个子节点

4. 求解子节点

计算子节点的LP松弛

- a. 若LP不可行,则返回第2步
- b. 若 $f^{LP} \leq f^*$,则返回第2步(本质上也是不可行)

5. 检查整数可行性

若LP松弛的解满足整数条件,更新 $f^* = f^{LP}$ (定界),返回第2步

6. 分支

选定一个取值不为整数的变量 x_i ,分支生成两个新节点加入N,返回第3步

■ 分支定界算法 – 进阶话题

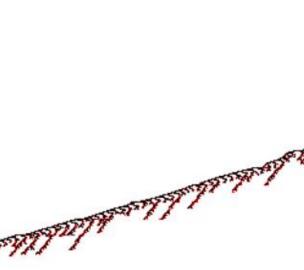


分支策略

- 随机选择
- 选择最接近整数取值的
- 选择最不接近整数取值的
- 打分分支策略 (例如基于目标函数值)

节点选择策略

- 深度优先
- 广度优先
- 最佳整数值提升优先

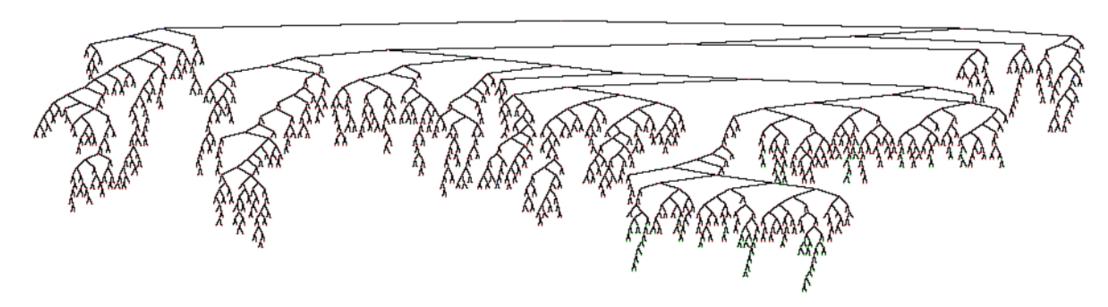


■ 分支定界算法 – 进阶话题



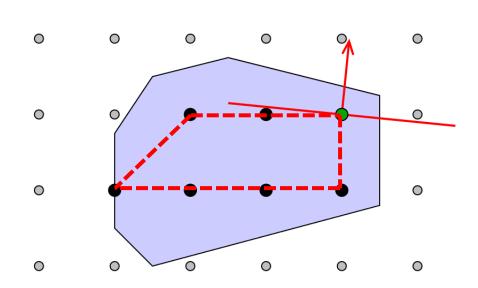
并行化

- 共享内存并行
 - 每次选择多个节点并行计算
 - 受单机计算能力、内存影响
- 分布式的并行
 - 切分为多个子任务
 - 超大规模下子任务间的同步和信息共享量较大

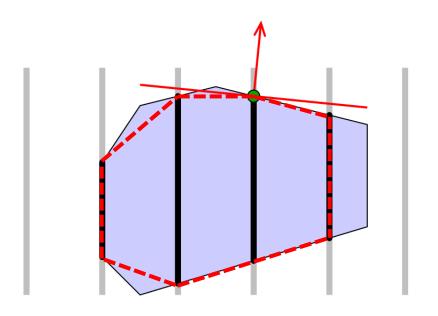


■ 割平面算法 - 凸包





整数规划 Integer programming (IP)



混合整数规划 Mixed-Integer programming (MIP)

■ 割平面算法 – 割平面

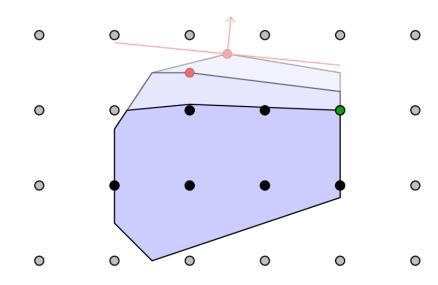


通用割平面算法

- Gomory
- Mixed-Integer Rounding (MIR)
- Zero-Half

针对问题结构的割平面算法

- Knapsack cover
- Clique
- Multi-commodity-flow (MCF)



■ 割平面算法 – Knapsack cover – 简介



什么是背包 (Knapsack)

$$\sum_{j \in N} a_j x_j \le b$$
$$a_j, b \in \mathbb{Z}_+, x \in \{0,1\}^n$$

例如:
$$5x_1 + 6x_2 + 2x_3 + 2x_4 \le 8$$

什么是背包的cover (Knapsack cover) *C* ⊆ N

$$\sum_{j \in C} a_j > b \quad \Rightarrow \quad \sum_{j \in C} x_j \le |C| - 1$$

■ 割平面算法 – Knapsack cover – 一个例子



例如

max
$$3x_1 + 5x_2 + 5x_3 + 4x_4 + x_5$$

s.t. $3x_1 + 6x_2 + 7x_3 + 6x_4 + 2x_5 \le 18$
 $x_1, x_2, x_3, x_4, x_5 \in \{0,1\}$

初始LP松弛解

$$x_1 = x_2 = x_3 = 1, x_4 = \frac{1}{3}, x_5 = 0$$

一个Cover

$$\{x_2, x_3, x_4\} \Rightarrow x_2 + x_3 + x_4 \le 2$$

新的LP松弛解

$$x_1 = x_2 = x_3 = 1, x_4 = 0, x_5 = 1$$

思考: 其他解法

1.
$$f = 14\frac{1}{3}$$

 $x_1 = x_2 = x_3 = 1, x_4 = \frac{1}{3}, x_5 = 0$

2.
$$x_4 = 0$$
, $f = 14$
 $x_1 = x_2 = x_3 = 1$, $x_4 = 0$, $x_5 = 1$

3.
$$x_4 = 1$$
, $f = 14\frac{1}{7}$
 $x_1 = x_2 = 1$, $x_3 = \frac{3}{7}$, $x_4 = 1$, $x_5 = 0$

■ 割平面算法 – Knapsack cover – 如何计算



1. 使用辅助变量 $z \in \{0,1\}^n$, $z_i = 1$ 当且仅当 $j \in N$ 构成一个cover, C, 因此可得

$$\sum\nolimits_{j\in N}z_j=|C|$$

2. 由cover的定义可得

$$\sum_{j \in N} a_j z_j \ge b + 1$$

3. 希望cover割平面 $\sum_{j \in C} x_j \le |C| - 1$ 可以割去当前的非整数LP解 x^* ,即 x^* 带入后不满足,可以写为

$$\sum_{j \in N} x_j^* z_j > |C| - 1 = \sum_{j \in N} z_j - 1 \Rightarrow \sum_{j \in N} (1 - x_j^*) z_j < 1$$

4. 得到优化问题

$$\min \sum_{j \in N} (1 - x_j^*) z_j$$

s. t. $\sum_{j \in N} a_j z_j \ge b + 1, z \in \{0,1\}^n$

5. 当上述优化问题的最优值小于1时,则找到一个cover,以及一个确定有效的 cover cut

■ 割平面算法 – 进阶话题



割平面处理

- 割平面提升 (cut lifting)
- 割平面强化

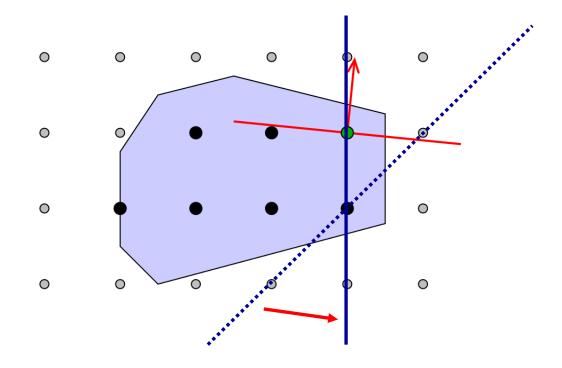
割平面选择

• 有效性: 割的越深越好

• 正交性: 和其他个平面尽量正交

• 平行性:和目标函数尽量平行

• 低秩性: 使用的已有割平面越少越好 (Low rank)



■ 实例 - 数独 - 数学模型



数独规则

每一行、每一列、每一个3×3的粗线格子内的数字均含1-9且不重复,给定部分初始赋值

Difficulty: Evil \checkmark 列 j=1...9 Auto-Check for Mistakes

					3			8
2		4		6			9	
	1							
	7						5	
5		1			6	3		
	9			1				
			2			9		
7								
6		5		4			2	

行 *i* = 1..9

数学建模

变量: 行i 列j 取值为v 时 $x_{ijv} = 1$,否则 $x_{ijv} = 0$ $x_{ijv} \in \{0,1\}, \quad 1 \le i,j,v \le 9$

约束:

每个空格均有一个赋值

$$\sum_{v=1}^{9} x_{ijv} = 1, \qquad 1 \le i, j \le 9$$

每一行均含1-9, 且不重复:

$$\sum_{j=1}^{9} x_{ijv} = 1, \qquad 1 \le i, v \le 9$$

每一列均含1-9, 且不重复:

$$\sum_{i=1}^{9} x_{ijv} = 1, \qquad 1 \le j, v \le 9$$

每一个 3×3 的粗线格子 $(1 \le s, t \le 3)$ 均含1-9,且不重复:

$$\sum_{i=3s-2}^{3s} \sum_{j=3t-2}^{3t} x_{ijv} = 1, \qquad 1 \le s, t \le 3, 1 \le v \le 9$$

部分初始赋值,当行i 列j 初始赋值为v 时

$$x_{ijv} = 1$$
例如 $x_{629} = 1$

■ 实例 - 数独 - 优化模型: 变量



数学建模

变量: 行i 列j 取值为v 时 $x_{ijv} = 1$,否则 $x_{ijv} = 0$

$$x_{ijv} \in \{0,1\}, \qquad 1 \le i, j, v \le 9$$

约束:

每个空格均有一个赋值

$$\sum_{v=1}^{9} x_{ijv} = 1, \qquad 1 \le i, j \le 9$$

每一行均含1-9,且不重复:

$$\sum_{j=1}^{9} x_{ijv} = 1, \qquad 1 \le i, v \le 9$$

每一列均含1-9, 且不重复:

$$\sum_{i=1}^{9} x_{ijv} = 1, \qquad 1 \le j, v \le 9$$

每一个 3×3 的粗线格子 $(1 \le s, t \le 3)$ 均含1-9,且不重复:

$$\sum_{i=3s-2}^{3s} \sum_{j=3t-2}^{3t} x_{ijv} = 1, \quad 1 \le s, t \le 3, 1 \le v \le 9$$

部分初始赋值, 当行i 列j 初始赋值为v 时

$$x_{ijv} = 1$$
 例如 $x_{629} = 1$

```
#!/usr/bin/python
import sys
import math
from coptpy import *

# Create model instance
env = Envr()
model = env.createModel('sudoku')

# Create variables
x = model.addVars(9, 9, 9, vtype=COPT.BINARY)
```

■ 实例 - 数独 - 优化模型: 约束



数学建模

变量: 行i 列j 取值为v 时 $x_{ijv} = 1$,否则 $x_{ijv} = 0$

$$x_{ijv} \in \{0,1\}, \qquad 1 \le i, j, v \le 9$$

约束:

每个空格均有一个赋值

$$\sum_{v=1}^{9} x_{ijv} = 1, \qquad 1 \le i, j \le 9$$

每一行均含1-9, 且不重复:

$$\sum_{j=1}^{9} x_{ijv} = 1, \qquad 1 \le i, v \le 9$$

每一列均含1-9, 且不重复:

$$\sum_{i=1}^{9} x_{ijv} = 1, \qquad 1 \le j, v \le 9$$

每一个 3×3 的粗线格子 $(1 \le s, t \le 3)$ 均含1-9,且不重复:

$$\sum_{i=3s-2}^{3s} \sum_{j=3t-2}^{3t} x_{ijv} = 1, \quad 1 \le s, t \le 3, 1 \le v \le 9$$

部分初始赋值, 当行i 列j 初始赋值为v 时

$$x_{iiv} = 1$$
例如 $x_{629} = 1$

```
# Each cell must take one value
model.addConstrs((x.sum(i, j, '*') == 1
                 for i in range(9)
                 for j in range(9)))
# Each value appears once per row
model.addConstrs((x.sum(i, '*', v) == 1
                 for i in range(9)
                 for v in range(9)))
# Each value appears once per column
model.addConstrs((x.sum('*', j, v) == 1
                 for j in range(9)
                 for v in range(9))
# Each value appears once per subgrid
model.addConstrs((
    quicksum(x[i, j, v] for i in range(3*s, 3*(s + 1))
                        for j in range(3*t, 3*(t + 1))) == 1
             for s in range(3)
             for t in range(3)
             for v in range(9)))
```

■ 实例 - 数独 - 优化模型: 初始赋值



数学建模

变量: 行i 列j 取值为v 时 $x_{ijv} = 1$,否则 $x_{ijv} = 0$

$$x_{ijv} \in \{0,1\}, \qquad 1 \le i, j, v \le 9$$

约束:

每个空格均有一个赋值

$$\sum_{v=1}^{9} x_{ijv} = 1, \qquad 1 \le i, j \le 9$$

每一行均含1-9,且不重复:

$$\sum_{j=1}^{9} x_{ijv} = 1, \qquad 1 \le i, v \le 9$$

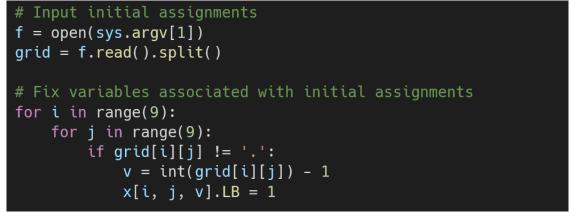
每一列均含1-9,且不重复:

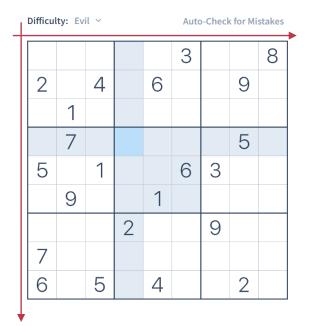
$$\sum_{i=1}^{9} x_{ijv} = 1, \qquad 1 \le j, v \le 9$$

每一个
$$3 \times 3$$
的粗线格子 $(1 \le s, t \le 3)$ 均含 $1-9$,且不重复:
$$\sum_{i=3s-2}^{3s} \sum_{j=3t-2}^{3t} x_{ijv} = 1, \qquad 1 \le s, t \le 3, 1 \le v \le 9$$

部分初始赋值,当行i 列j 初始赋值为v 时

$$x_{ijv} = 1$$
 例如 $x_{629} = 1$





输入文件内容:
38
2.4.69.
.1
.75.
5.163
.91
29
7
6.5.42.

■ 实例 - 数独 - 优化模型: 求解和输出

```
# Solve the model
model.solve()

print('')
print('Solution:')
print('')

# Retrieve optimization result
solution = model.getInfo(COPT.Info.Value, x)

for i in range(9):
    sol = ''
    for j in range(9):
        if solution[i, j, v] == 1:
            sol += str(v + 1)
    print(sol)
```

备注:使用了杉数求解器COPT求解https://shanshu.ai/copt

© 2022 Cardinal Operations PowerPoint Business Theme. All Rights Reserved.

```
Cardinal Optimizer v4.0.4. Build date Mar 22 2022
Copyright Cardinal Operations 2022. All Rights Reserved
Hardware has 4 cores and 8 threads. Using instruction set X86_AVX2 (10)
Minimizing a MIP problem
The original problem has:
    324 rows, 729 columns and 2916 non-zero elements
    729 binaries
Presolving the problem
The presolved problem has:
    0 rows, 0 columns and 0 non-zero elements
Best solution : 0.000000000
Best bound
               : 0.000000000
Best gap
               : 0.0000%
Solve time
               : 0.01
Solve node
               : 0
MIP status
              : solved
Solution status : integer optimal
Violations
                     absolute
                                  relative
  bounds
                                         0
  rows
  integrality :
```

Solution:

967523418 254861793

813974265 476382159

581496372 392715684 148257936

729638541 635149827

■ 实例 - 数独 - 优化模型: 完整建模部分

数学建模

变量: 行i 列j 取值为v 时 $x_{ijv} = 1$,否则 $x_{ijv} = 0$

$$x_{ijv} \in \{0,1\}, \quad 1 \le i, j, v \le 9$$

约束:

每个空格均有一个赋值

$$\sum_{v=1}^{9} x_{ijv} = 1, \qquad 1 \le i, j \le 9$$

每一行均含1-9, 且不重复:

$$\sum_{i=1}^{9} x_{ijv} = 1, \qquad 1 \le i, v \le 9$$

每一列均含1-9, 且不重复:

$$\sum_{i=1}^{9} x_{ijv} = 1, \qquad 1 \le j, v \le 9$$

每一个 3×3 的粗线格子 $(1 \le s, t \le 3)$ 均含1-9,且不重复:

$$\sum_{i=3s-2}^{3s} \sum_{j=3t-2}^{3t} x_{ijv} = 1, \quad 1 \le s, t \le 3, 1 \le v \le 9$$

部分初始赋值, 当行i 列j 初始赋值为v 时

$$x_{ijv} = 1$$
例如 $x_{629} = 1$

```
# Create variables
x = model.addVars(9, 9, 9, vtype=COPT.BINARY)
# Each cell must take one value
model.addConstrs((x.sum(i, j, '*') == 1
                 for i in range(9)
                 for j in range(9)))
# Each value appears once per row
model.addConstrs((x.sum(i, '*', v) == 1
                 for i in range(9)
                 for v in range(9)))
# Each value appears once per column
model.addConstrs((x.sum('*', j, v) == 1
                 for j in range(9)
                 for v in range(9)))
# Each value appears once per subgrid
model.addConstrs((
    quicksum(x[i, j, v] for i in range(3*s, 3*(s + 1))
                        for j in range(3*t, 3*(t + 1))) == 1
             for s in range(3)
             for t in range(3)
             for v in range(9)))
# Input initial assignments
f = open(sys.argv[1])
grid = f.read().split()
# Fix variables associated with initial assignments
for i in range(9):
    for j in range(9):
        if grid[i][j] != '.':
            v = int(grid[i][j]) - 1
            x[i, j, v].LB = 1
```

思考: 如何验证数独的唯一性?

■ 拓展话题 – 整数规划能解多大的问题



rail02



Track allocation problem modeled as arc coupling problem

tokyometro

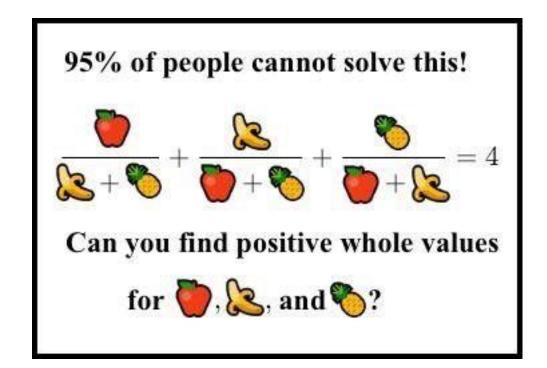


The layout model for Tokyo Metro Map

参考网站: http://miplib.zib.de/

■ 拓展话题 - 求解器遇到的数值问题





(非线性) 整数规划

$$min a + b + p$$

s.t.
$$\frac{a}{b+p} + \frac{b}{a+p} + \frac{p}{a+b} = 4$$
$$a, b, p \in N$$

为何难以正确求解?

a = 4373612677928697257861252602371390152816537558161613618621437993378423467772036

b = 36875131794129999827197811565225474825492979968971970996283137471637224634055579

p = 154476802108746166441951315019919837485664325669565431700026634898253202035277999

■ 拓展话题 – 数值范围较大带来的问题



•
$$(a + b) - b = ?$$

•
$$(10^{-7} + 10^{-4}) - 10^{-4} = 1.0000000000000 \times 10^{-7}$$

•
$$(10^{-7} + 10^{-3}) - 10^{-3} = 1.000000000001 \times 10^{-7}$$

•
$$(10^{-7} + 10^4) - 10^4 = 1.000007614493 \times 10^{-7}$$

•
$$(10^{-7} + 10^7) - 10^7 = 1.005828380585 \times 10^{-7}$$

•
$$(10^{-7} + 10^8) - 10^8 = 1.043081283569 \times 10^{-7}$$

•
$$(10^{-7} + 10^9) - 10^9 = 1.192092895508 \times 10^{-7}$$

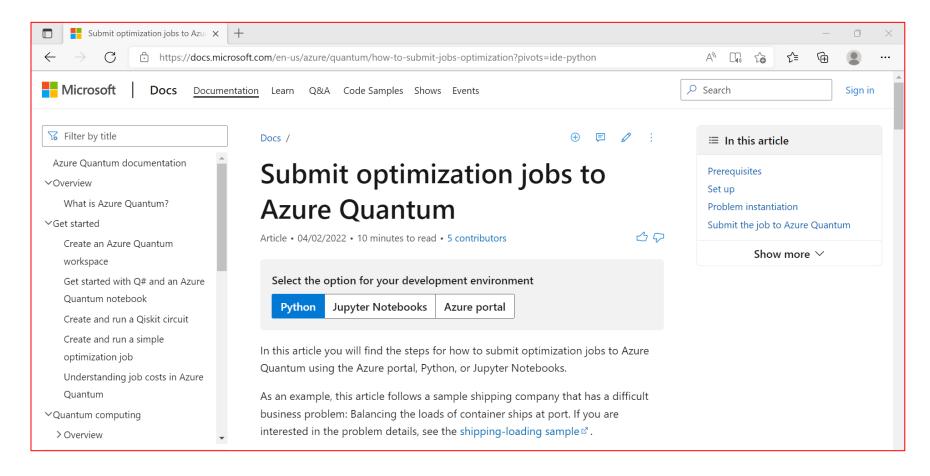
•
$$(10^{-7} + 10^{10}) - 10^{10} = 0$$

■ 拓展话题 – 量子计算



量子计算离用的到还有多远量子计算在整数规划的进展

"Most existing approaches to quantum optimization are intended to solve unconstrained binary programming problems"





THANKS -



& 400-680-5680



Shanshu@shanshu.ai