清華大学 Tsinghua University

运筹学

第六讲 混合整数线性规划

魏韡

2025年 4月23日

主要内容

1.1 混合整数规划的数学模型

$$\begin{aligned} \min\quad & c^{\mathrm{T}}x + d^{\mathrm{T}}y \\ \text{s.t.}\quad & Ax + By \le b \\ & x \in Z^m, y \in R^n \end{aligned}$$

$$\begin{aligned} \min\quad & c^{\mathrm{T}}x + d^{\mathrm{T}}y \\ \text{s.t.}\quad & Ax + By \le b \\ & x \in \{0,1\}^m, y \in R^{n-m} \end{aligned}$$

混合整数线性规划

- x: 整数变量
- y: 连续变量
- 目标函数是线性函数
- 约束是线性等式与不等式

0-1整数线性规划

- 整数变量取 0 或 1

$$x = x^L + z_1 + z_2 + \cdots + z_{x^U - x^L}$$

$$x = x^L + z_1 + 2z_2 + \cdots + 2^{N-1} z_N$$

$$N = \lceil \log_2 (x^U - x^L) \rceil$$

1.2 机组组合问题

<u>**问题描述**</u>: 求发电机组次日 T 个时段内的启停状态，使得可以通过调整在线机组出力满足负荷曲线，且运行成本最低

- 网络约束机组组合：考虑潮流约束
- 安全约束机组组合：考虑 $N-k$ 故障

min 总运行成本
s.t. 最小开机时间约束
最小停机时间约束
机组发电容量约束
机组爬坡约束
备用容量约束
网络潮流模型
传输线安全约束

决策变量
• 整数变量 u_{it}: 机组 i 在 t 时段的状态，运行($u_{it}=1$)/停机($u_{it}=0$)
• 连续变量 p_{it}: 机组 i 在 t 时段的输出功率
• 潮流模型中的其他变量及辅助变量

1.2 机组组合问题

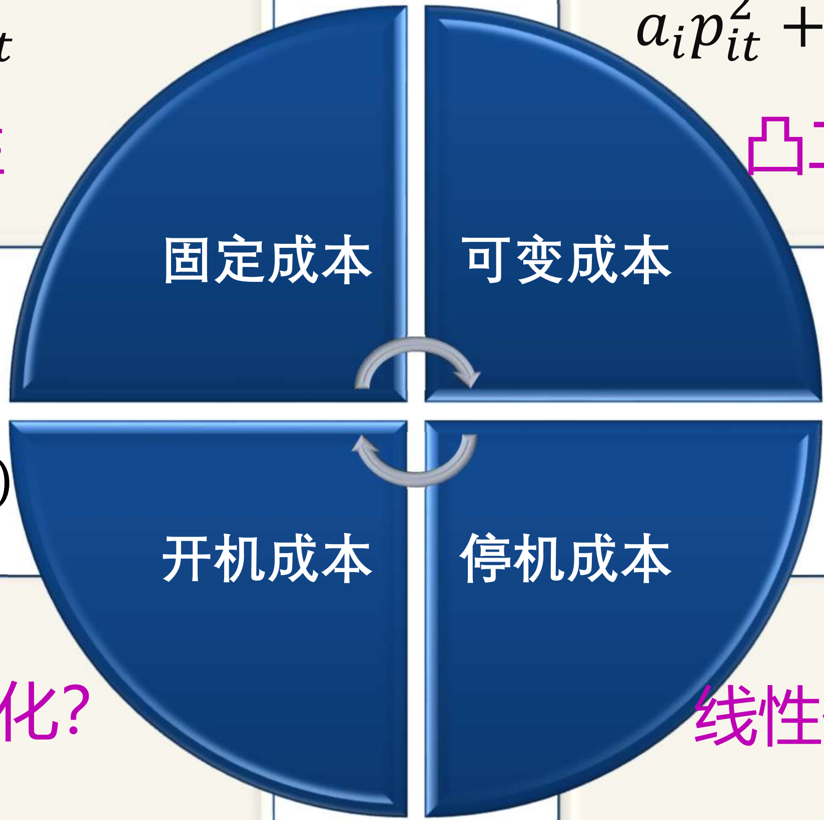
目标函数 $\min \sum_i \sum_t \left(\sigma_{it}^s + \sigma_{it}^d + F_i u_{it} + f_i(p_{it})\right)$

$F_i u_{it}$

线性

$a_i p_{it}^2 + b_i p_{it}$

凸二次



σ_{it}^s: 连续变量

$\sigma_{it}^s \rightarrow \min$: $\sigma_{it}^s \geq 0$

$\sigma_{it}^s \geq S_i^U (u_{it} - u_{it-1})$

σ_{it}^d: 连续变量

$\sigma_{it}^d \rightarrow \min$: $\sigma_{it}^d \geq 0$

$\sigma_{it}^d \geq S_i^D (u_{it-1} - u_{it})$

线性化?

线性化?

$S_i^U (1 - u_{it-1}) u_{it}$

$S_i^D u_{it-1} (1 - u_{it})$

1.2 机组组合问题

最小连续开、停机时间约束:

机组开启/关停后，至少连续运行 T_{on}/T_{off}时段

$$u_{ik} \geq u_{it} - u_{it-1},\ k = t, t+1, \cdots, \min\{t + T_{on} - 1, T\}$$

$$u_{ik} \leq 1 - u_{it-1} + u_{it},\ k = t, t+1, \cdots, \min\{t + T_{off} - 1, T\}$$

机组发电容量约束:

哪种表示更好?

$$u_t = 0 \Rightarrow p_t = 0$$

$$u_t = 1 \Rightarrow p_l \leq p_t \leq p_m$$

$$\begin{cases} p_t = u_t(p_l + \eta_t(p_m - p_l)),\ \eta \in [0,1] \\ u_t p_l \leq p_t \leq u_t p_m \end{cases}$$

机组爬坡约束:

在两个连续时段内机组出力增减受爬坡率R_i^+/R_i^-限制

$$p_{it} - p_{it-1} \leq R_i^+$$

$$p_{it-1} - p_{it} \leq R_i^-$$

启停过程中机组出力增减量必然不小于p_i^l，可能导致机组组合无可行解

线路潮流约束: $-F_{ij}^l \leq \sum_i \pi_{il} p_{it} - \sum_j \pi_{jl} d_{jt} \leq F_{ij}$

1.2 机组组合问题

$$\begin{aligned}
\min\ & \sum_t \sum_t \left[SU_{it} + SD_{it} + F_{it}u_{it} + f(p_{it})\right] \\
\text{s.t.}\ & SU_{it} \ge 0,\ SU_{it} \ge S_i^U(u_{it} - u_{it-1}), \forall i, \forall t \\
& SD_{it} \ge 0,\ SD_{it} \ge S_i^D(u_{it-1} - u_{it}), \forall i, \forall t \\
& u_{it} - u_{it-1} - u_{ik} \le 0,\ \forall i, \forall t, k = t : \min\{t + T_{on} - 1, T\} \\
& u_{it-1} - u_{it} + u_{ik} \le 1,\ \forall i, \forall t, k = t : \min\{t + T_{off} - 1, T\} \\
& \sum_i p_{it} = \sum_j d_{jt},\ \forall t
\end{aligned}$$

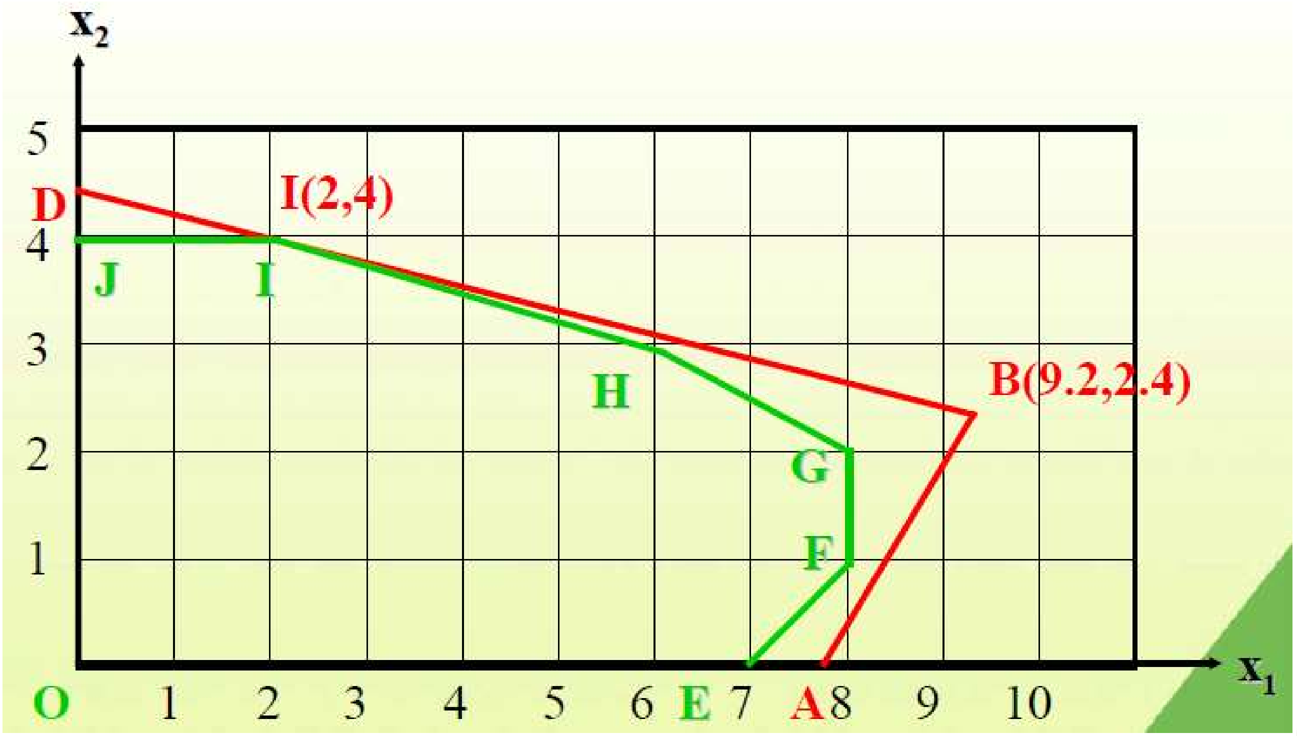
$$\begin{aligned}
& \sum_i u_{it} p_i^l \le (1-r) d_i,\ \sum_i u_{it} p_i^m \ge (1+r) d_i,\ \ \forall t \\
& p_{it} \le p_{it-1} + R_i^+, \forall i, \forall t \\
& p_{it-1} \le p_{it} + R_i^-, \forall i, \forall t \\
& u_{it} p_i^l \le p_{it} \le u_{it} p_i^m, \forall i, \forall t \\
& -F_{ij} \le \sum_i \pi_{il} p_{it} - \sum_j \pi_{jl} d_{jt} \le F_{ij}, \forall l, \forall t
\end{aligned}$$

yalmip 程序 https://yalmip.github.io/example/unitcommitment/

1.3 整数可行域非凸性带来的困难

按连续变量求解线性规划后就近取整如何?

$$\max\ 3x_1 + 13x_2$$
$$s.t.\ 2x_1 + 9x_2 \le 40$$
$$11x_1 - 8x_2 \le 82$$
$$x_1, x_2 \in Z_+$$



	LP松弛	就近取整	最近整数	最优解
(x_1, x_2)	(9.2, 2.4)	(9, 2)	(8, 2)	(2, 4)
目标值	58.8	不可行	50	58

1. 数学模型与机组组合问题
2. 分支定界算法
3. 线性化方法
4. 例题

2.1 混合整数线性规划计算能力的提升

- CPLEX 1.2 (1991) → CPLEX 11 (2007): 提升29000倍
- Gurobi 1.0 (2009) 与 CPLEX 11 相当
- Gurobi 1.0 → Gurobi 6.5 (2015): 提升48.7倍
- 1991-2015: 提升140万倍
- 在30年前的老式计算机上用 CPLEX1.2 需要**16天**才能求解的 MILP 问题用 Gurobi 6.5 在相同的计算机上用 **1 秒**即可求解.
- 1993 → 2016: 处理器速度提升 160万倍
- 总效率提升: 2.2万亿倍
- 在30年前的老式计算机上用 CPLEX1.2 需要 **71000年**才能求解的 MILP 问题用 Gurobi 6.5 在2016年的计算机上只需 **1秒** 即可求解.
- **尽管 MILP 是非凸优化问题, 但通常能在合理的时间内求解**

Bertsimas D. Machine Learning under a Modern Optimization Lens.
https://www.lnmb.nl/conferences/2018/programlnmbconference/Bertsimas-1.pdf

2.1 混合整数线性规划计算能力的提升

没有任何一个算法在最坏情况下都表现良好

广泛使用的方法有两类

- **割平面算法**. 通常用于纯整数规划，对于整数凸优化（连续松弛后的问题是凸优化问题）也适用，需要修改割平面生成方式。
- **分支定界算法**. 亦可用于混合整数线性规划
- 二者的结合. 好的割平面能够被分支定界法的子问题继承从而提高搜索效率，在现代求解器中广泛采用
- 机器学习加速求解策略

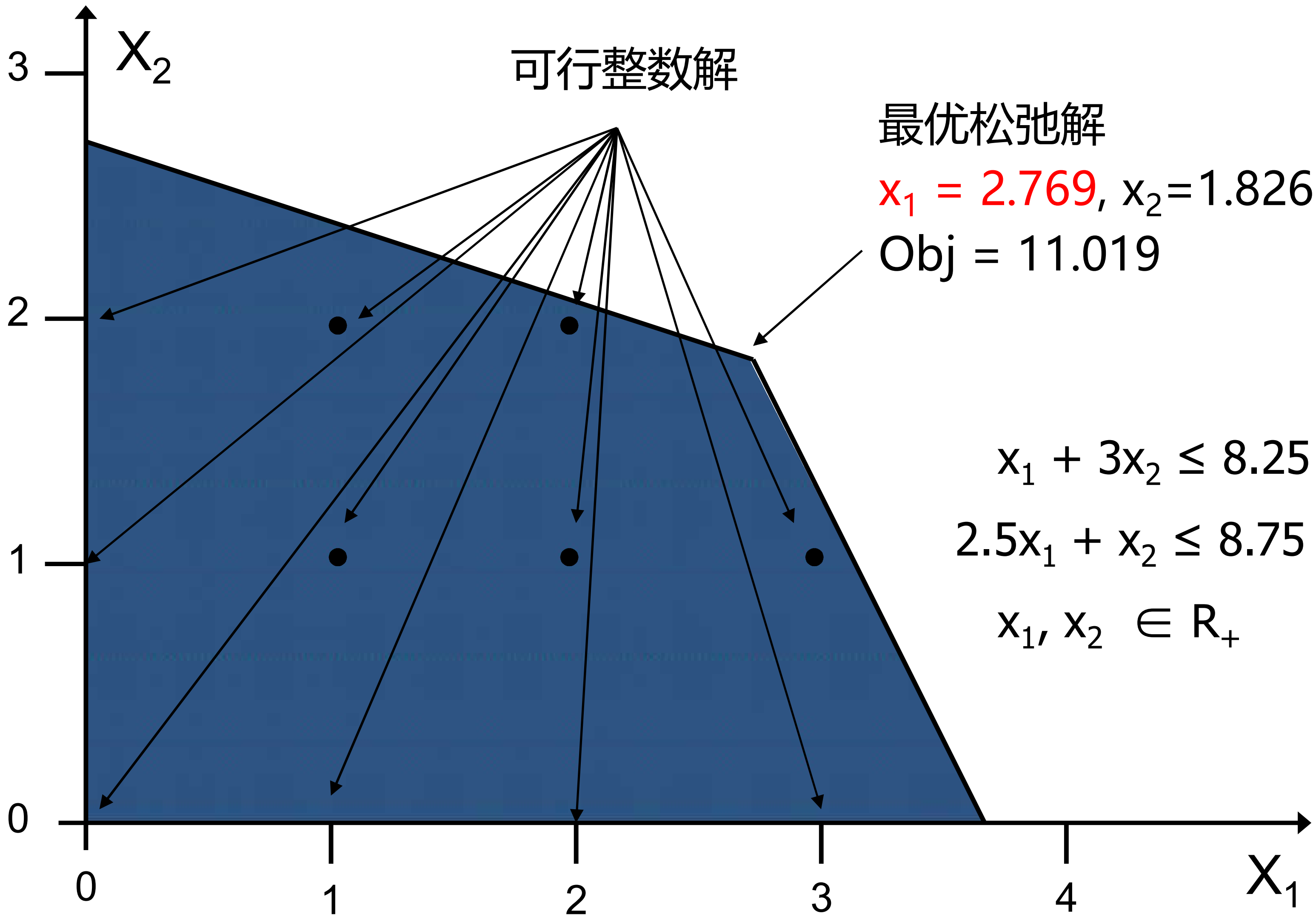
2.2 分支定界算法

$$\begin{aligned} \max \quad & 2x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + 3x_2 \le 8.25 \\ & 2.5x_1 + x_2 \le 8.75 \\ & x_1, x_2 \in Z_+ \end{aligned}$$

凸松弛 ⇓ (线性规划)

$$\begin{aligned} \max \quad & 2x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + 3x_2 \le 8.25 \\ & 2.5x_1 + x_2 \le 8.75 \\ & x_1, x_2 \in R_+ \end{aligned}$$

2.2 分支定界算法



3
2
1
0
X_2
0
1
2
3
4
X_1
可行整数解
最优松弛解
x_1 = 2.769, x_2=1.826
Obj = 11.019
$x_1 + 3x_2 \le 8.25$
$2.5x_1 + x_2 \le 8.75$
$x_1, x_2 \in R_+$

2.2 分支定界算法

对变量 x_1 分支

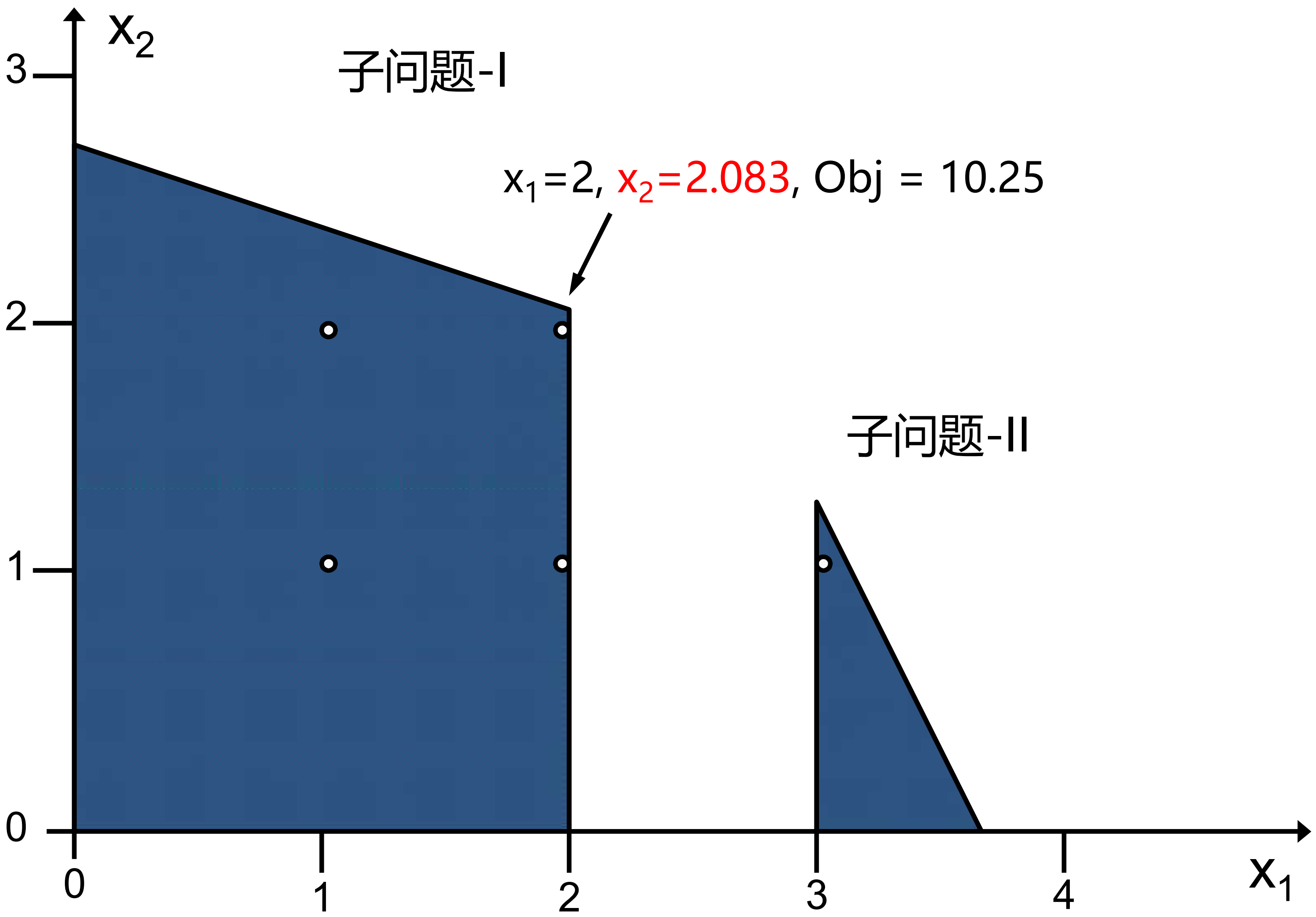
子问题-I:

$$
\begin{aligned}
\max \quad & 2x_1 + 3x_2 \\
\text{s.t.} \quad & x_1 + 3x_2 \le 8.25 \\
& 2.5x_1 + x_2 \le 8.75 \\
& \boxed{x_1 \le 2} \\
& x_1, x_2 \in R_+
\end{aligned}
$$

子问题-II:

$$
\begin{aligned}
\max \quad & 2x_1 + 3x_2 \\
\text{s.t.} \quad & x_1 + 3x_2 \le 8.25 \\
& 2.5x_1 + x_2 \le 8.75 \\
& \boxed{x_1 \ge 3} \\
& x_1, x_2 \in R_+
\end{aligned}
$$

2.2 分支定界算法



2.2 分支定界算法

进一步划分子问题-I的可行域，对变量 x_2 分支

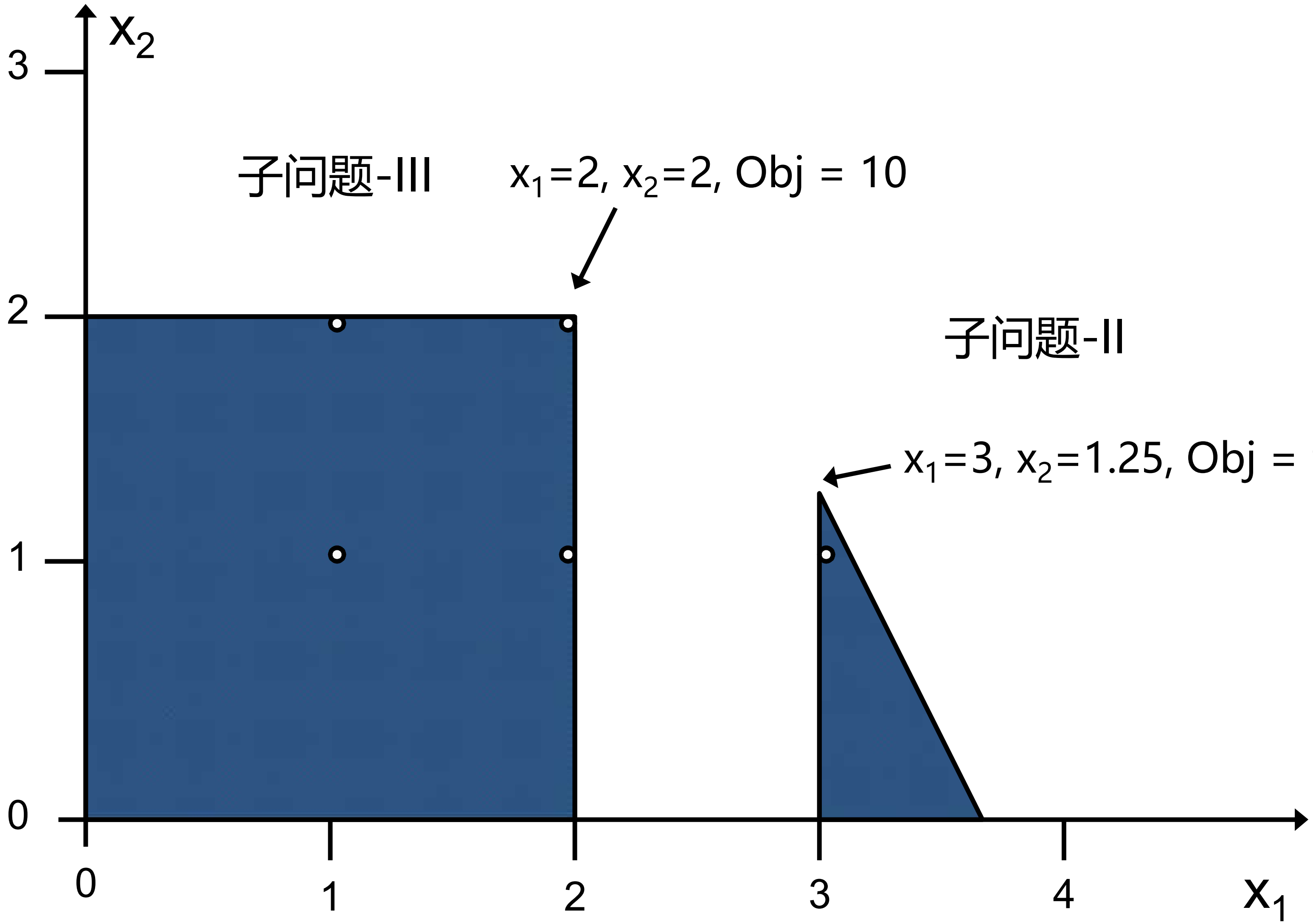
子问题-III

$$
\begin{aligned}
\max \quad & 2x_1 + 3x_2 \\
\text{s.t.} \quad & x_1 + 3x_2 \le 8.25 \\
& 2.5x_1 + x_2 \le 8.75 \\
& x_1 \le 2 \\
& x_2 \le 2 \\
& x_1, x_2 \in R_+
\end{aligned}
$$

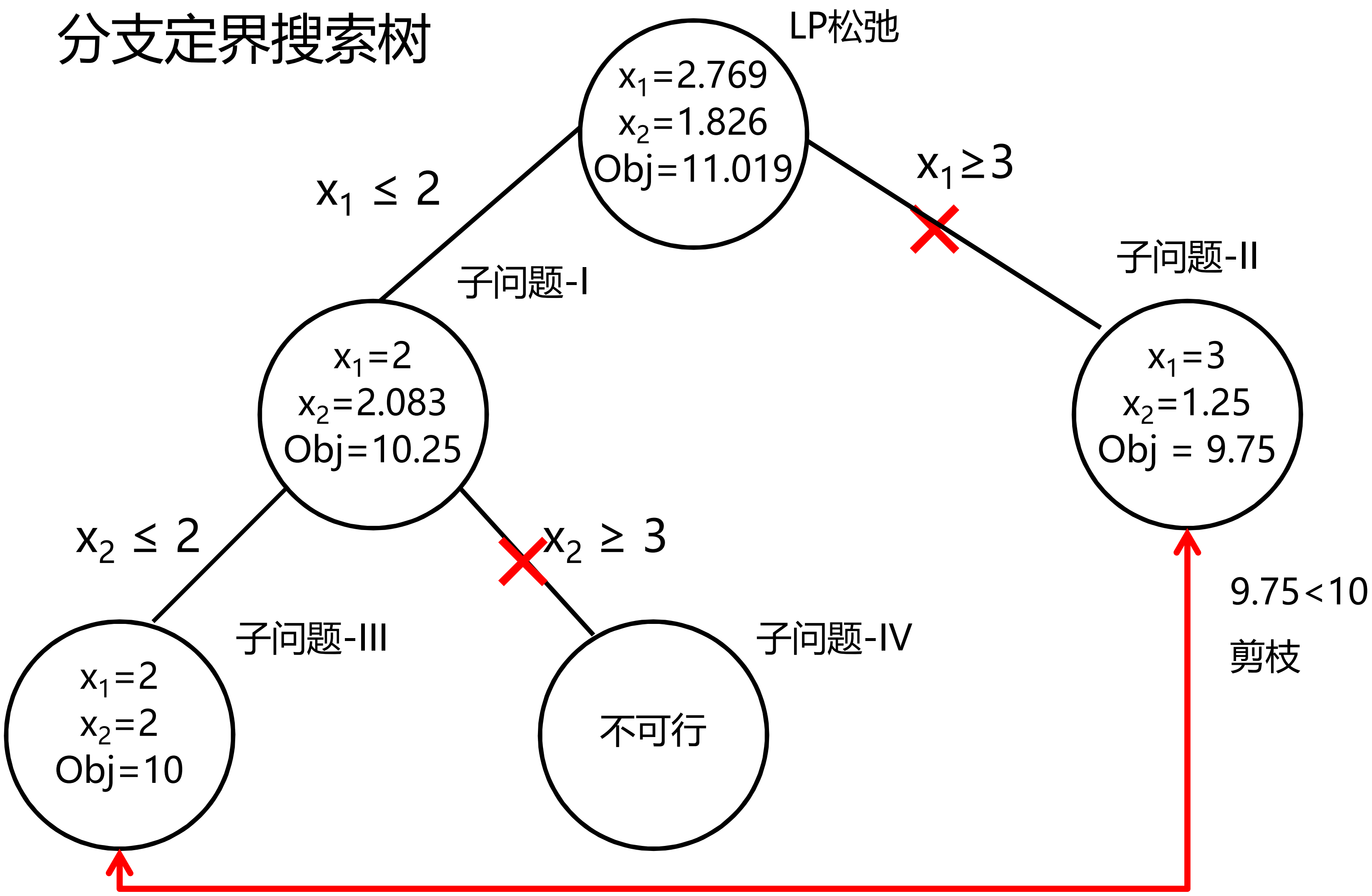
子问题-IV

$$
\begin{aligned}
\max \quad & 2x_1 + 3x_2 \\
\text{s.t.} \quad & x_1 + 3x_2 \le 8.25 \\
& 2.5x_1 + x_2 \le 8.75 \\
& x_1 \le 2 \\
& x_2 \ge 3 \\
& x_1, x_2 \in R_+
\end{aligned}
$$

2.2 分支定界算法



2.2 分支定界算法


分支定界搜索树
LP松弛
x1=2.769
x2=1.826
Obj=11.019
x1 ≤ 2
x1≥3
子问题-I
子问题-II
x1=2
x2=2.083
Obj=10.25
x1=3
x2=1.25
Obj = 9.75
x2 ≤ 2
x2 ≥ 3
9.75<10
剪枝
子问题-III
子问题-IV
x1=2
x2=2
Obj=10
不可行

2.3 分支定界算法小结（极小化问题）

算法流程

设 x_f 是已知的可行整数解，z_f 是该可行解对应的目标函数值

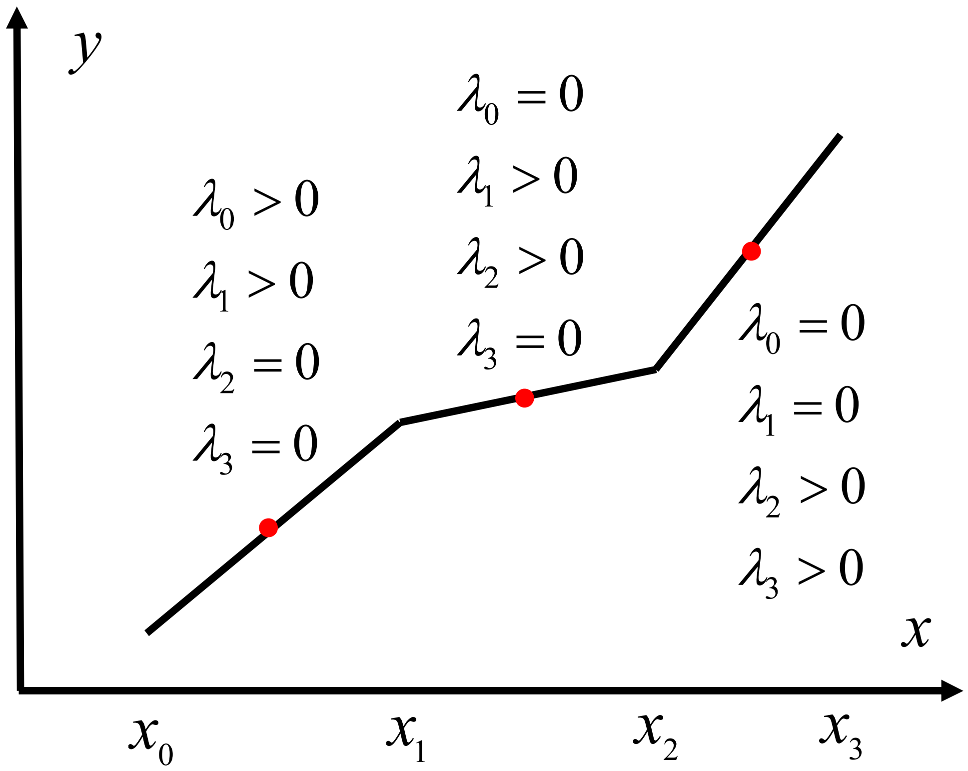
1. 将原问题的松弛可行域记为节点 (1)，将该节点激活
2. 若存在被激活的节点，按照某个规则选择一个节点 (i)
3. 求解节点(i)对应的线性规划，得到 $\left(x_{LP}^{(i)}, z_{LP}^{(i)}\right)$
   - 若 $z_{LP}^{(i)} \geq z_f$，删除节点(i)，即剪枝
   - 若 $z_{LP}^{(i)} < z_f$ 且 $x_{LP}^{(i)}$ 是整数，删除节点(i)并更新当前最好可行解和最优值上界 $\left(x_f, z_f\right) \leftarrow \left(x_{LP}^{(i)}, z_{LP}^{(i)}\right)$
   - 若 $z_{LP}^{(i)} < z_f$ 且 $x_{LP}^{(i)}$ 不是整数，删除节点(i)的同时对某个不整变量分支，创建两个新的子问题（子节点）
4. 回到第2步

2.3 分支定界算法小结（极小化问题）

- **基本思想**: 分支定解法是一种针对整数变量的分类讨论策略，分类过程构成一个分支树保证不重不漏。
- **定下界**: 求解若干松弛问题 (原问题或分支树上子问题的松弛问题)，松弛问题是线性规划或其他容易求解的问题，最优值是原问题或子问题的下界。
- **定上界**: 保存当前已知最好的整数可行解，对应的目标函数是最优值的一个上界。若在求解过程中发现了更好的可行解则更新上界，因此上界是单调下降的序列。
- **剪枝**: 若某个分支上，松弛问题的最优解大于最优上界，该分支上所有可行的整数解对应的目标函数值不会更小，因此该分支必不包含最优解，删除该分支，也不再搜索该分支的可行域。
- **收敛性**：分支过程在子问题中不断添加新的约束，分支越深下界越大，上界单调下降；每次分支后整体可行域变小；算法有限次分支后必定收敛，但最坏情况下复杂度指数增长。

3.1 单变量连续函数

表示 一个非凸分段线性函数



定义

称向量 $\lambda \in \mathbb{R}_+^n, n \geq 2$ 为类型2特殊序集，若 λ 中至多有两个连续的元素为正数，其余元素都是0

$$x = \Sigma_i \lambda_i x_i$$

$$y = \Sigma_i \lambda_i f(x_i)$$

$$\lambda \geq 0,\ \Sigma_i \lambda_i = 1$$

$$\lambda \text{ is SOS2}$$

3.1 单变量连续函数

类型2特殊序集的实现

可以建模为整数规划 $x \in (x_1, x_2) \Rightarrow \lambda_{1,2} > 0, \lambda_{0,3,\cdots,n} = 0$

$$\begin{aligned}
&\lambda_0 \le \theta_1 \\
&\lambda_1 \le \theta_1 + \theta_2 \\
&\lambda_2 \le \theta_2 + \theta_3 \\
&\quad \vdots \\
&\lambda_{k-1} \le \theta_{k-1} + \theta_k \\
&\lambda_k \le \theta_k \\
&\lambda_i \ge 0, i = 1, \cdots, k, \textstyle\sum_{i=0}^{k} \lambda_i = 1 \\
&\theta_s \in \{0,1\}, s = 1, \cdots, k, \textstyle\sum_{s=1}^{k} \theta_s = 1
\end{aligned}$$

$$\begin{aligned}
&0 \le \lambda_0 \le 0 \\
&0 \le \lambda_1 \le 1 \\
&0 \le \lambda_2 \le 1 \\
&\quad \vdots \\
&0 \le \lambda_{k-1} \le 0 \\
&0 \le \lambda_k \le 0 \\
&\lambda_1 + \lambda_2 = 1 \\
&\theta_2 = 1, \theta_{1,3,\cdots,n} = 0
\end{aligned}$$

3.2 乘积项

两个0-1整数变量的乘积

$$z = xy,\ x, y \in \{0,1\} \Leftrightarrow \begin{array}{l} 0 \le z \le y \\ 0 \le x - z \le 1 - y \end{array}$$ ✓

检验:

$$y = 0 \Rightarrow \begin{cases} 0 \le z \le 0 \\ 0 \le x - z \le 1 \end{cases} \Rightarrow \begin{cases} z = 0 \\ 0 \le x \le 1 \end{cases}$$ ✓

$$x = 0 \Rightarrow \begin{cases} 0 \le z \le y \\ 0 \le 0 - z \le 1 - y \end{cases} \Rightarrow \begin{cases} z = 0 \\ 0 \le y \le 1 \end{cases}$$ ✓

$$\begin{cases} x = 1 \\ y = 1 \end{cases} \Rightarrow \begin{cases} 0 \le z \le 1 \\ 0 \le 1 - z \le 0 \end{cases} \Rightarrow z = 1$$ ✓

如何线性化两个一般整数变量的乘积?

Gupte A, Ahmed S, Cheon M S, et al. Solving Mixed Integer Bilinear Problems Using MILP Formulations. SIAM Journal on Optimization, 2011, 23(2):721-744.

3.2 乘积项

连续变量与0-1整数变量的乘积

$$z = xy,\ x \in [x_l, x_m], y \in \{0,1\} \Leftrightarrow \begin{matrix} x_l y \le z \le x_m y \\ x_l(1-y) \le x - z \le x_m(1-y) \end{matrix}$$

检验:

$$y = 0 \Rightarrow \begin{cases} 0 \le z \le 0 \\ x_l \le x \le x_m \end{cases} \Rightarrow \begin{cases} z = 0 \\ x_l \le x \le x_m \end{cases}$$

$$y = 1 \Rightarrow \begin{cases} x_l \le z \le x_m \\ 0 \le x - z \le 0 \end{cases} \Rightarrow \begin{cases} x_l \le z \le x_m \\ x = z \end{cases}$$

3.2 乘积项

0-1整数变量构成的单项式

$$z = x_1 x_2 \cdots x_n,\quad x_i \in \{0,1\},\ i = 1,\cdots,n$$

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

只要有一个 x_i 为 0，约束右端项必定小于$(n-1)/n$，从而将$z=1$排除在可行域外，因此 $z=0$ 是唯一可能的情况

$$z \le \frac{x_1 + x_2 + \cdots + x_n}{n}$$

若所有 x_i 都等于1, 该约束右端项为$1/n$, 从而将$z=0$排除在可行域外，因此 $z=1$ 是唯一可能的情况

$$z \ge \frac{x_1 + x_2 + \cdots + x_n - n + 1}{n}$$

3.3 其他常用表达

多个变量取极小值

$$y = \min\{x_1, x_2, \cdots, x_n\},\quad x_i \in [x_i^l, x_i^m],\ L = \min\{x_1^l, \cdots, x_n^l\}$$

$$x_i^l \le x_i \le x_i^m,\ \forall i,$$

$$y \le x_i, \forall i$$

$$\boxed{x_i - (x_i^m - L)(1 - z_i) \le y, \forall i}$$

$$z_i \in \{0,1\},\quad \textstyle\sum_{i=1}^n z_i = 1$$

$z_i = 1$ 对最小的 x_i

$z_i = 0$ 对其他 $x_i - y \le x_i^m - L$

z 没有其他可行的整数解

若 x_1 最小, $z_3 = 1$ 对 $x_3 > x_1$,

则 $y \le x_1$ 与 $x_3 \le y$ 矛盾

3.3 其他常用表达

多个变量取极大值

$$y = \max\{x_1, x_2, \cdots, x_n\}, \quad x_i \in [x_i^l, x_i^m], \; U = \max\{x_1^m, \cdots, x_n^m\}$$

$$x_i^l \le x_i \le x_i^m, \; \forall i,$$

$$y \ge x_i, \forall i$$

$$\boxed{x_i + (U - x_i^l)(1 - z_i) \ge y, \forall i}$$

$$z_i \in \{0,1\}, \quad \sum_{i=1}^{n} z_i = 1$$

$z_i = 1$ 对最大的 x_i

$z_i = 0$ 对其他 $y - x_i \le U - x_i^l$

z 没有其他可行的整数解

若 x_1 最大，但 $z_3 = 1$ 且 $x_3 < x_1$，

则 $y \ge x_1$ 与 $x_3 \ge y$ 矛盾

3.3 其他常用表达

互补松弛条件

$$0 \le x \perp y \ge 0$$

$$\Updownarrow$$

$$x, y \ge 0, x^{\mathrm{T}} y = 0$$

$$\Updownarrow$$

$$x, y \ge 0, x_i y_i = 0, \forall i$$

$$\begin{aligned} &0 \le x \le Mz \\ &0 \le y \le M(1-z) \quad ✓ \\ &z \in \{0,1\}^n \end{aligned}$$

检验:

$$x_i > 0 \Rightarrow z_i = 1 \Rightarrow y_i = 0 \quad ✓$$

$$y_i > 0 \Rightarrow z_i = 0 \Rightarrow x_i = 0 \quad ✓$$

3.3 其他常用表达

逻辑条件

- a 与 b：$c = ab$ $\{0,1\} \times \{0,1\}$
- a_1 与 a_2 与 ... a_n: c = $a_1 a_2 \cdots a_n$ 0-1单项式
- a 或 b：$c = \max\{a, b\}$ $a \le c \le 1,\ b \le c \le a + b$
- a_1 或 a_2 或 ... a_n: $c = \max\{a_1, a_2, \cdots, a_n\}$

$$a_i \le c, \forall i \quad c \le 1 \quad c \le \sum_{i=1}^{n} a_i$$

主要内容

4. 例题

例1

- 变量 x 在区间 $[l_1, u_1]$ 或区间 $[l_2, u_2]$ 中，其中 $u_1 < l_2$。将该约束表示为线性不等式

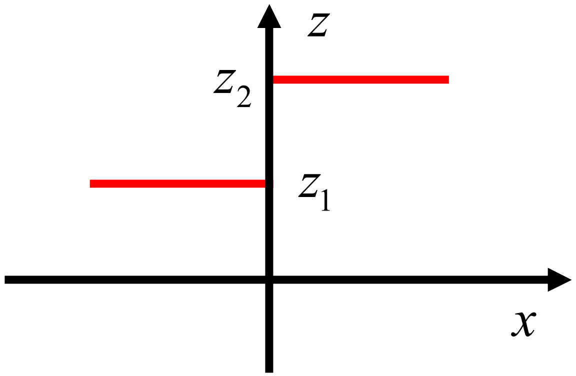
$$l_1 \le x \le u_1 \quad \text{or} \quad l_2 \le x \le u_2$$

$$l_1 y + l_2(1-y) \le x \le u_1 y + u_2(1-y)$$

例2

- 表示阶跃函数

$$z = \begin{cases} z_1, & x < 0 \\ z_2, & x > 0 \end{cases}$$



$$|x| \le x_M$$

$$z = z_1 y_1 + z_2 y_2$$

$$y_1, y_2 \in \{0,1\}$$

$$y_1 + y_2 = 1$$

$$y_1 \ge -x/x_M$$

$$y_2 \ge x/x_M$$

当 $x = 0$ 时 z 取何值?

例3 置信区间：给定1000个采样数据 $x_1, x_2, \cdots, x_{1000}$。建立一个混合整数线性规划求长度最短的区间能够覆盖住至少900个采样点。

变量: $a, b \in \mathbb{R}$, $y_i = \begin{cases} 1, \text{若 } x_i \text{ 在区间 } [a,b] \text{ 中} \\ 0, \text{若 } x_i \text{ 在区间 } [a,b] \text{ 外} \end{cases}, i = 1,2,\cdots,1000$

判断点是否位于区间中

$$a \le x_i + M(1 - y_i), \forall i$$
$$x_i \le b + M(1 - y_i), \forall i$$
$$M = \max_i\{x_i\} - \min_i\{x_i\}$$

覆盖率约束

$$\sum_{i=1}^{1000} y_i \ge 900$$

区间边界

$$b \ge a$$

MILP
$\min \quad b - a$
$\text{s.t.} \quad b \ge a$
$a \le x_i + M(1 - y_i), \forall i$
$x_i \le b + M(1 - y_i), \forall i$
$y_i \in \{0,1\}, \forall i, \sum_{i=1}^{1000} y_i \ge 900$

例4 数独游戏

号称“史上最难”的数独谜题发表在2012年6月30日的《每日邮报》上。建立一个混合整数线性规划问题求出该数独的解

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

芬兰数学家Arto Inkala 花了3个月设计出一道数独谜题，它的解是唯一的。 Inkala声称这是有史以来最具挑战的数独谜题，只有最聪明的人才能解出来。

网格的81个方格中只有23个是已知的。谜题的难度取决于填满一个方格所需的逻辑推理次数。根据已知条件，很难直接推出某些方格里的数。

https://abcnews.go.com/blogs/headlines/2012/06/can-you-solve-the-hardest-ever-sudoku

例4　数独游戏

变量: $x_{ijk}=\begin{cases}1, 若\ k\ 在\ (i,j)\ 位置 \\ 0, \qquad 否则\end{cases}$

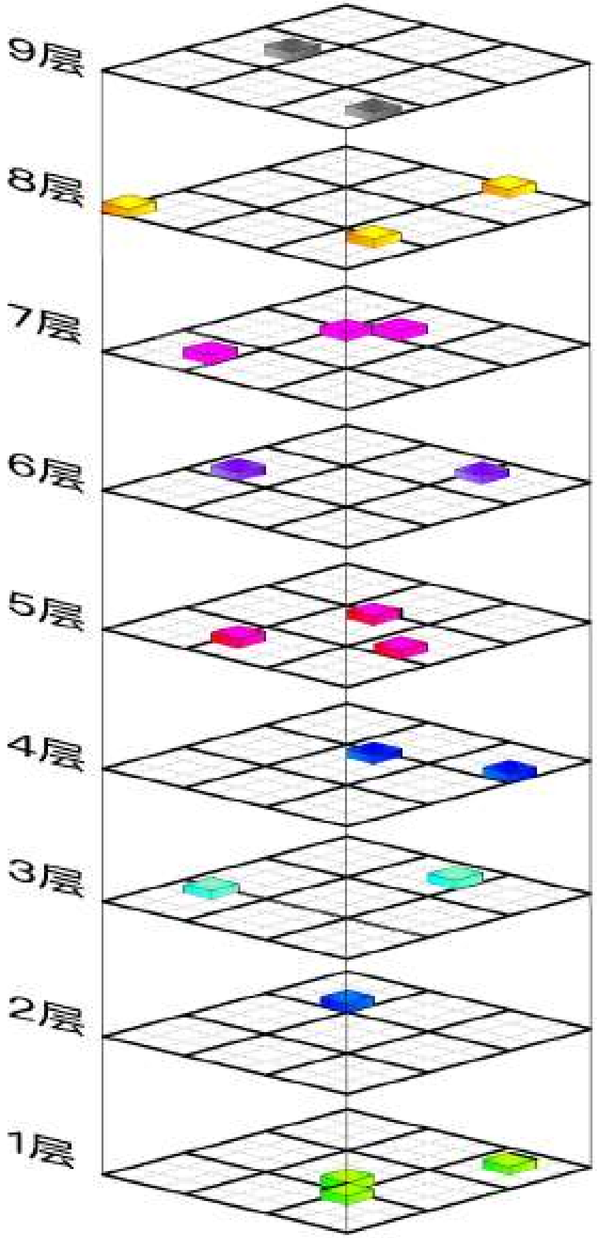
求满足以下约束条件的可行解

位置互斥: $\sum_{k=1}^{9} x_{ijk}=1, \forall i, \forall j$

行互斥: $\sum_{j=1}^{9} x_{ijk}=1, \forall i, \forall k$

列互斥: $\sum_{i=1}^{9} x_{ijk}=1, \forall j, \forall k$

已知条件: $x_{ijk}=1,$ 对给定的(i,j,k)



例4 数独游戏

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

初始盘面

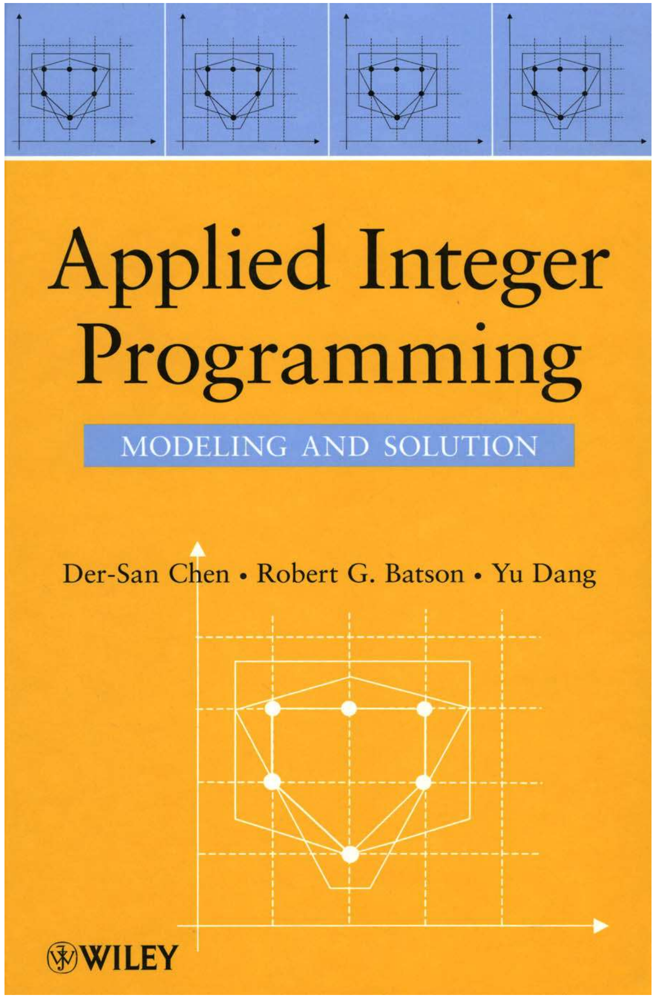
8	1	2	7	5	3	6	4	9
9	4	3	6	8	2	1	7	5
6	7	5	4	9	1	2	8	3
1	5	4	2	3	7	8	9	6
3	6	9	8	4	5	7	2	1
2	8	7	1	6	9	5	3	4
5	2	1	9	7	4	3	6	8
4	3	8	5	2	6	9	1	7
7	9	6	3	1	8	4	5	2

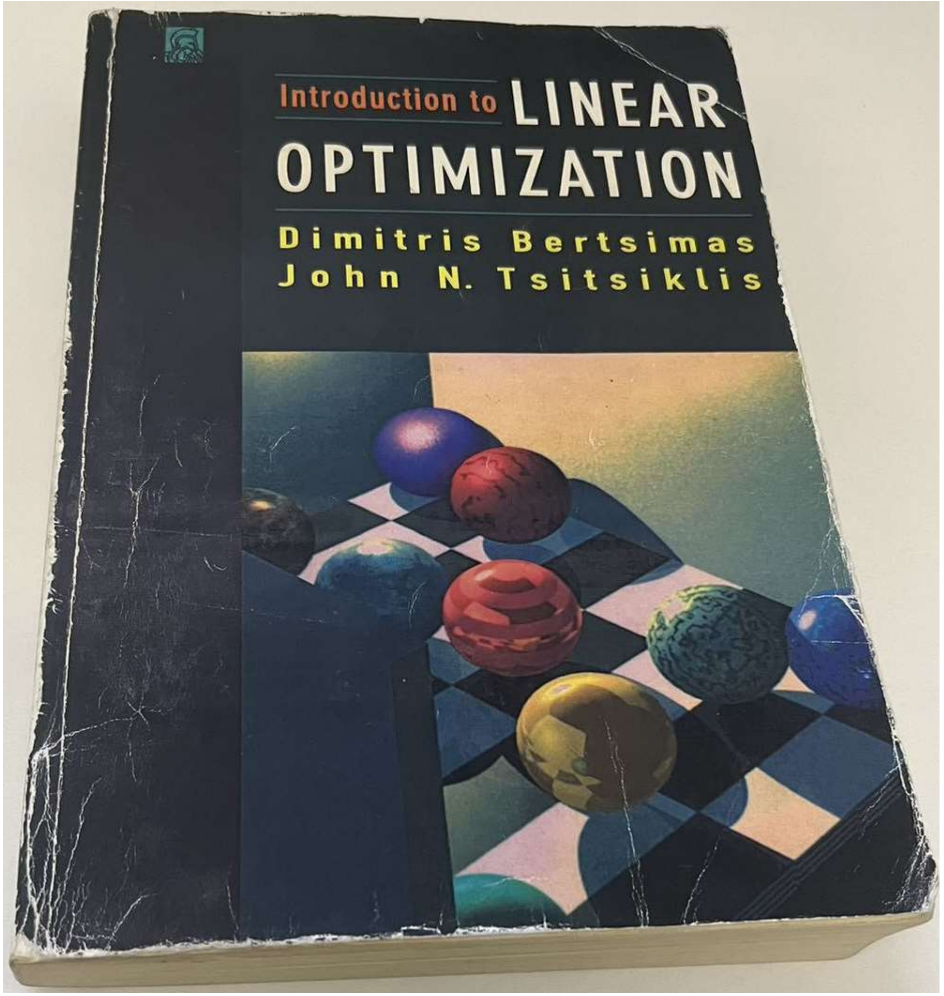
最终结果

思考

- 非线性规划总可以近似为MILP，但求解效率是否一定能提升?
- 什么情况下能提升?
- MILP属非凸优化，这种转化好处是什么?
- 等价的MILP模型是唯一的吗?
- 采用不同形式的MILP求解效率有多大差异?

推荐书目





Chen D S, Batson R G, Dang Y. Applied Integer Programming: Modeling and Solution. John Wiley & Sons, 2011.

Bertsimas D, Tsitsiklis J. Introduction to Linear Optimization. Athena Scientific, 1997