華東師範大學
EAST CHINA NORMAL UNIVERSITY

数据科学与工程算法基础

Algorithm Foundations of Data Science and Engineering

第十一章 整数规划

课程提纲

Content

课程提纲

Content

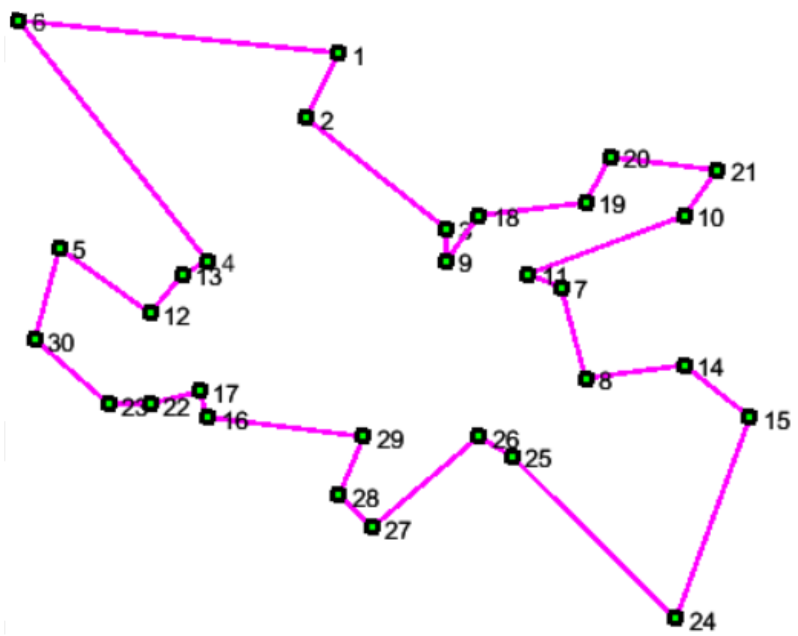
1 算法引入

2 分支定界法

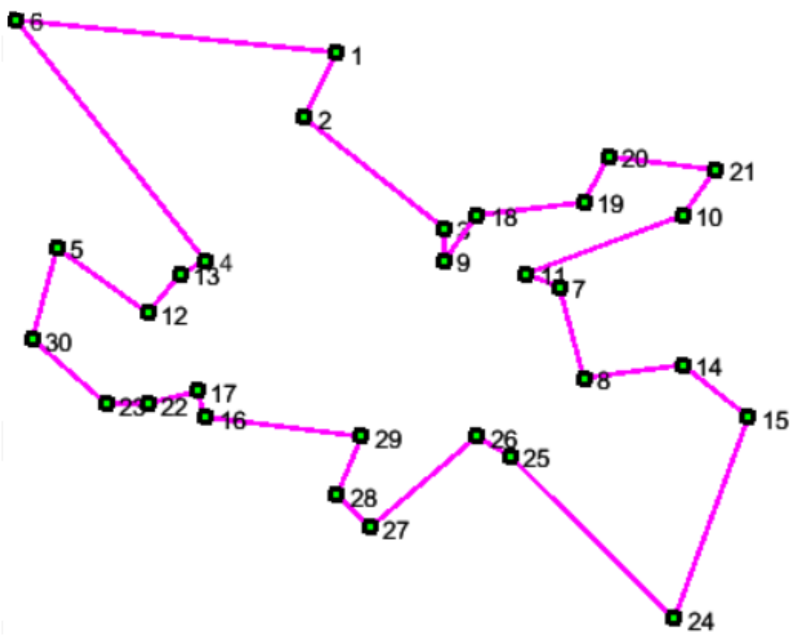
3 割平面法

旅行商问题（TSP）

- 给定一组城市以及每两个城市间的旅行成本（或距离），旅行商问题（TSP）旨在
  - 寻找一种遍访所有城市的最佳方案
  - 使得每个城市仅访问一次，而且旅行成本最小



旅行商问题（TSP）



- **输入**：包含 n 个点的集合，以及两两间的成本
- **可行方案**：如果遍访所有顶点，对对称 TSP 问题的可行解决方案为 $\dfrac{(n-1)!}{2}$
- **目标函数**：最小化旅行成本（或距离）的大小

TSP 问题可以广泛应用于各类调度问题：

车辆调度、物流调度、飞机调度、磁盘调度等等

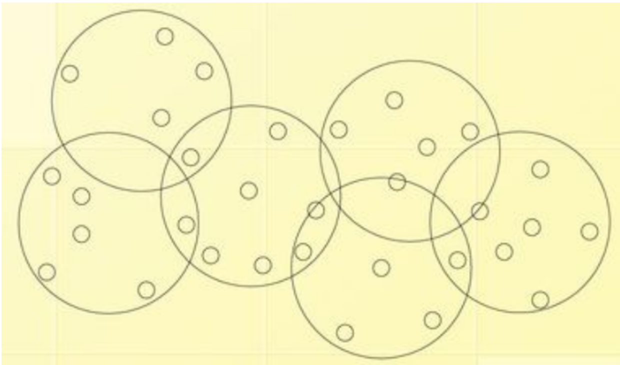
TSP 问题定义

- TSP 可以定义在一个无向图上
  - $G=(V,E)$，其中 $V=\{1,\ldots,n\}$ 为顶点集合
  - $E \in V \times V$ 为边的集合，每条边有成本标记 c_{ij}
- 定义 $x_{ij}=\begin{cases}1, & \text{if } e_{ij} \text{ is selected} \\ 0, & \text{otherwise}\end{cases}$，TSP 可以正式定义为

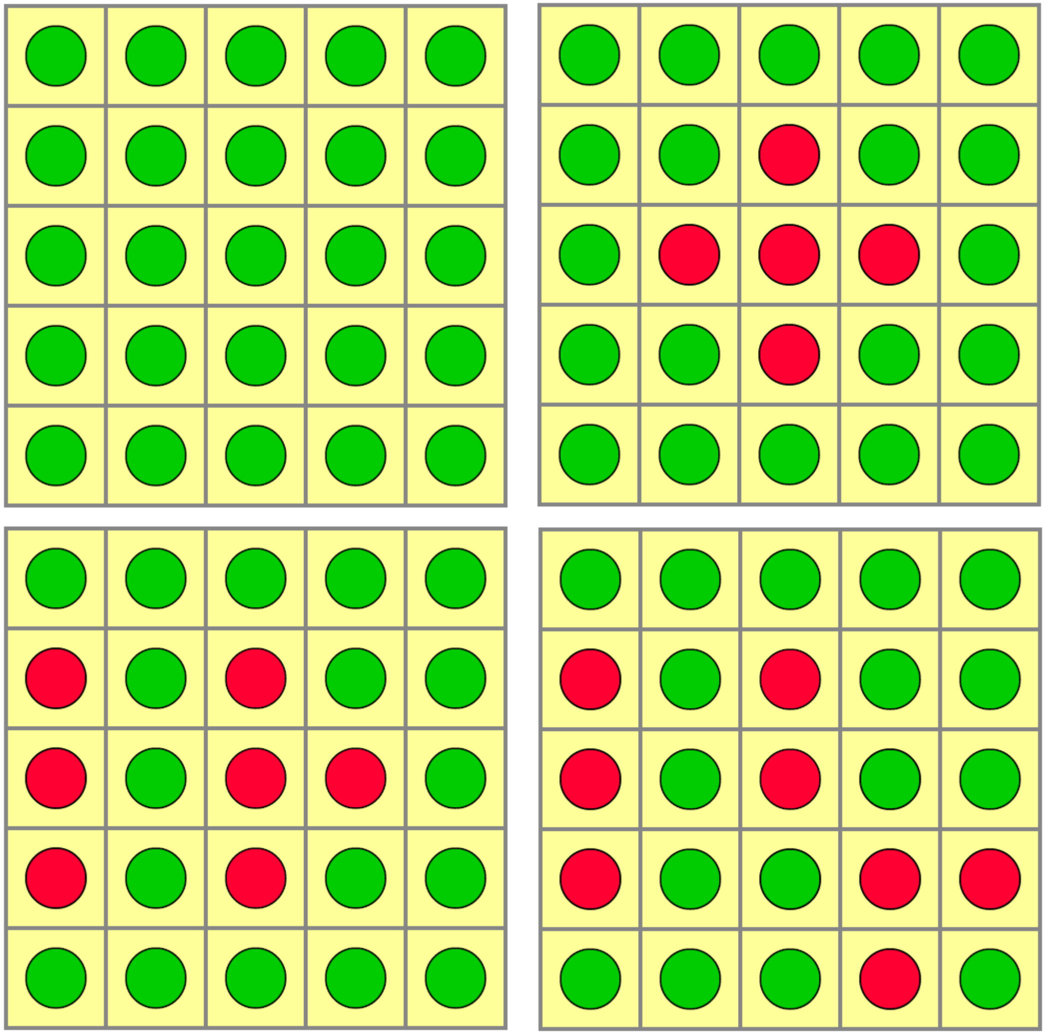
$$
\begin{aligned}
\min \quad & \sum_{i \neq j} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = 1,\ i \in V,\ i \neq j \\
& \sum_{i=1}^{n} x_{ij} = 1,\ j \in V,\ j \neq i \\
& \sum_{i,j \in S} x_{ij} \leq |S| - 1,\ S \subset V,\ 2 \leq |S| \leq n-2 \\
& x_{ij} \in \{0,1\},\ \forall i, j \in V
\end{aligned}
$$

集合覆盖问题（SCP）

- 输入
  - 全集 $U = \{u_1, u_2, \ldots, u_n\}$
  - 子集簇 $S = \{S_i \mid S_i \subseteq U, i \in [1,\ldots,m]\}$
  - 成本 $C = \{c_1, \ldots, c_m\}$
- 目标：找到一个指标集 $I \in \{1,\ldots,m\}$，使得 $\sum_{i \in I} c_i$ 最小且满足 $\bigcup_{i \in I} S_i = U$
- 集合覆盖问题具有广泛的应用
  - 有两大类问题：全覆盖和最大覆盖
  - 顶点覆盖
  - 信息传播
  - 文本摘要

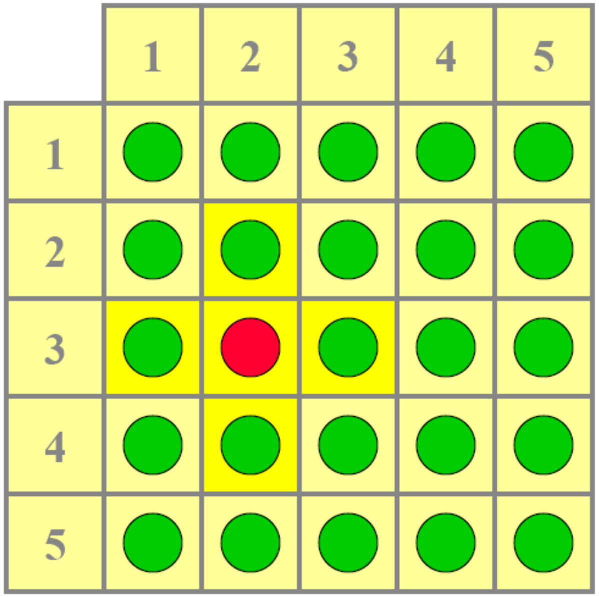


Fiver 游戏



- 游戏规则
  - 每个圆圈有红绿两种颜色
  - 单击一个圆圈后，圆圈及其相邻位置的圆圈颜色会翻转
  - 如图为分别点击 (3,3), (3,1) 和 (4,4) 后的结果
  - 如何以最少的点击次数，使所有圆圈都变成红色?

问题定义



- 注意到，同一个圆圈被点击两次，又恢复了原样
- 定义 $x_{ij} = \begin{cases} 1, & \text{circle } (i,j) \text{ is clicked} \\ 0, & \text{otherwise} \end{cases}$

$$\min \sum_{i=1}^{5}\sum_{j=1}^{5} x_{ij}$$

$$\text{s.t. } x_{ij} + x_{i(j-1)} + x_{i(j+1)} + x_{(i-1)j} + x_{(i+1)j} \text{ is odd}$$

$$x_{ij} \in \{0,1\}, \forall i,j \in [1,5]$$

$$x_{ij} = 0, \text{ otherwise}$$

问题定义

$$\min \sum_{i=1}^{5}\sum_{j=1}^{5} x_{ij}$$

$$\text{s.t. } x_{ij} + x_{i(j-1)} + x_{i(j+1)} + x_{(i-1)j} + x_{(i+1)j} - 2y_{ij} = 1$$

$$x_{ij} \in \{0,1\}, \forall i, j \in [1,5]$$

$$x_{ij} = 0, \text{ otherwise}$$

$$0 \le y_{ij} \le 2,\ y_{ij} \in \mathbb{N},\ 1 \le i, j \le 5$$

线性整数规划

- 输入：整数变量 $x_1, \ldots, x_n$ 和一组线性等式或不等式约束
- 可行解：满足所有不等式和等式以及完整性要求的变量组合
- 目标函数：$\max f(x_1, \ldots, x_n)$ 或者 $\min f(x_1, \ldots, x_n)$
- 示例

$$
\begin{aligned}
\min\ & 360x_1 + 400x_2 \\
\text{s.t. } & 20x_1 + 40x_2 \geq 180 \\
& 20x_1 + 10x_2 \geq 110 \\
& 0 \leq x_1, x_2 \in \mathbb{Z}
\end{aligned}
$$

20 TVs + 20 laundries

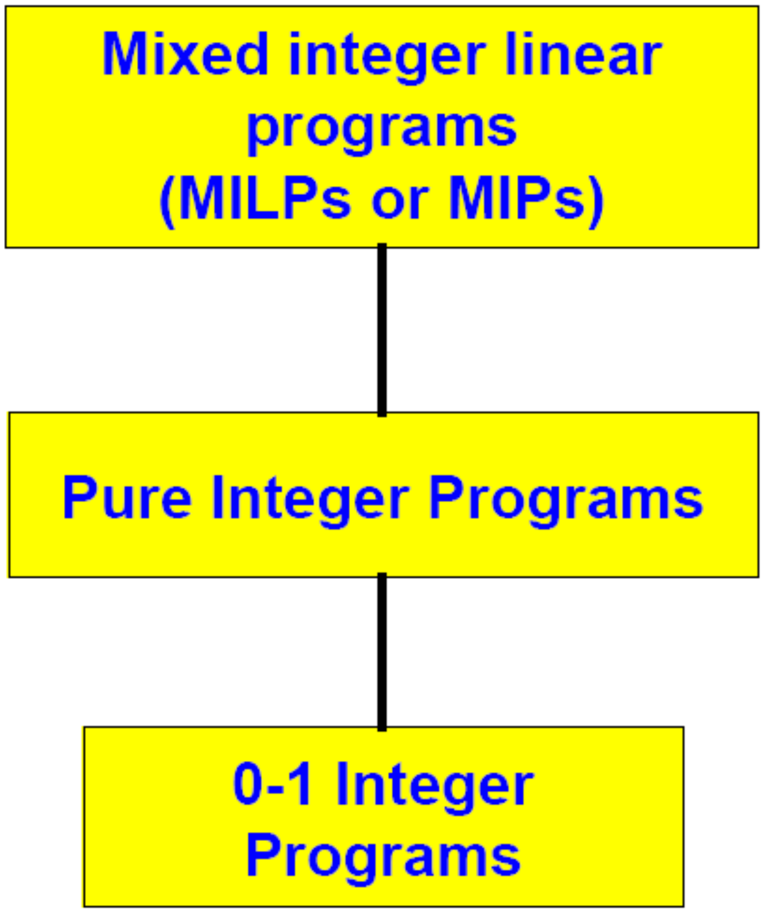
Cost: 360

40 TVs + 10 laundries

Cost: 400

需要运输 180 台电视和 110 台洗衣机

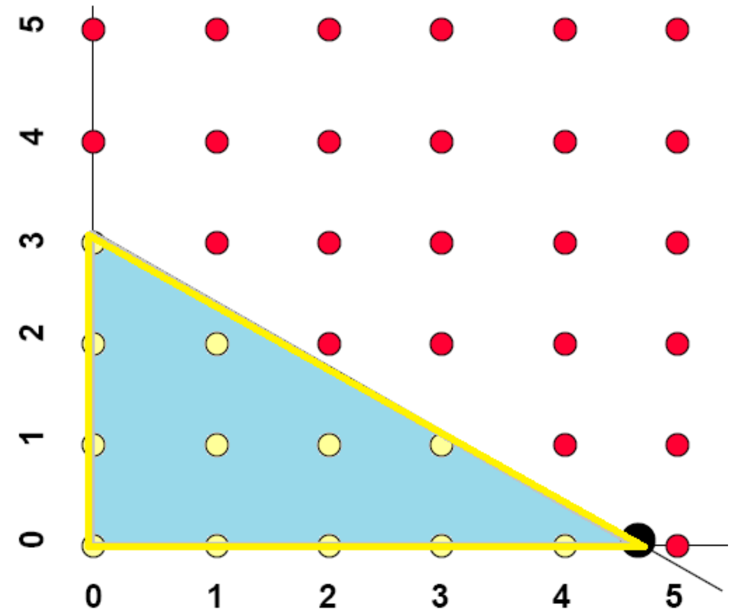
线性整数规划类型



- 混合线性整数规划
- 线性整数规划
- 0–1 线性规划

可行域

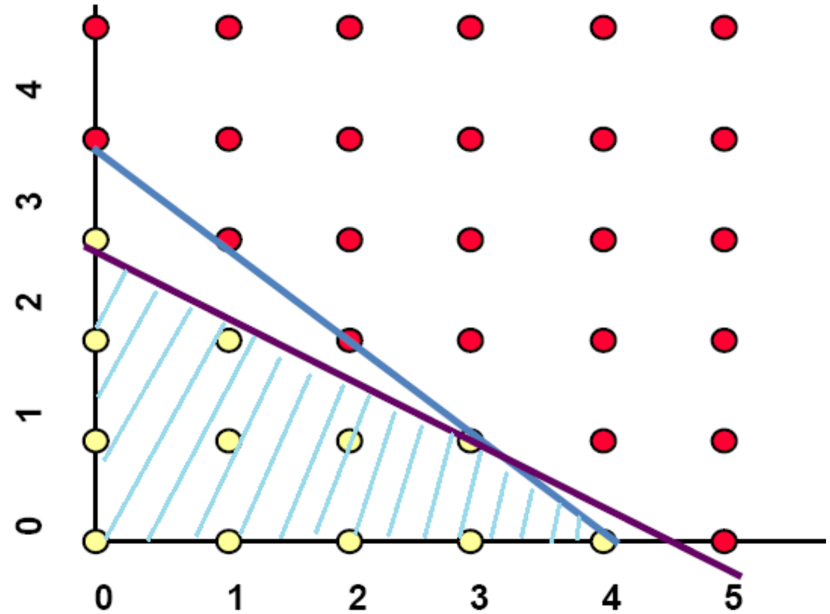
$$\max\ z = 3x + 4y$$
$$\text{s.t. } 5x + 8y \le 24$$
$$x, y \in \mathbb{Z}^+$$



- 该问题的解是什么?
- 可以使用线性规划来求解该问题吗?
  - 放弃整数约束，解得：$x = 4.8, y = 0, z = 14.4$
  - 不是整数解，向上取整 $x = 5, y = 0$，不在可行域内
  - 向下取整，$x = 4, y = 0, z = 12$，也不行
  - 最优解为 $x = 3, y = 1, z = 13$

可行域（续）

$$
\begin{aligned}
\max\ z &= 3x + 4y \\
\text{s.t. } x + y &\le 4 \\
2x + 3y &\le 9 \\
x, y &\in \mathbb{Z}^{+}
\end{aligned}
$$



- 更多的约束会减小可行区域的大小
- 求解过程中搜索空间将减少
- 但并不意味着求解问题会变得简单
- 通常，比解决线性规划问题要困难得多

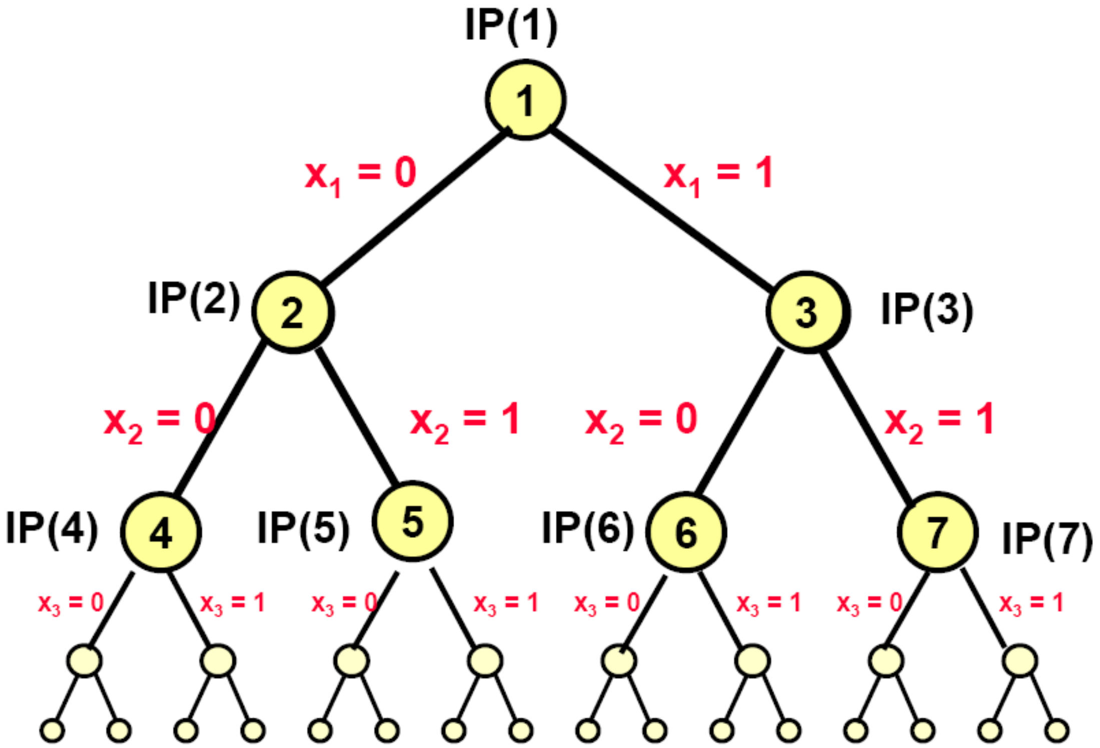
课程提纲

Content

枚举树

$$\max\ 24x_1 + 2x_2 + 20x_3 + 4x_4$$
$$\text{s.t. } 8x_1 + x_2 + 5x_3 + 4x_4 \le 9$$
$$x_i \in \{0,1\}, 1 \le i \le 4$$

- 背包问题的枚举树
  - 枚举树中有 31 个节点
  - 每个节点有两个孩子
  - 树的每个节点表示不同的整数规划问题



计算开销

- 假设每秒可以评估10亿个节点的最优解

n=30	1 sec.	n=60	31 years
n=40	17 minutes	n=70	31,000 years
n=50	11.6 days		

- 假设每秒可以评估 1 万亿个节点的最优解，并从枚举树中过滤掉 99.9999999% 的节点

n=70	1 sec.	n=100	31 years
n=80	17 minutes	n=110	31,000 years
n=90	11.6 days		

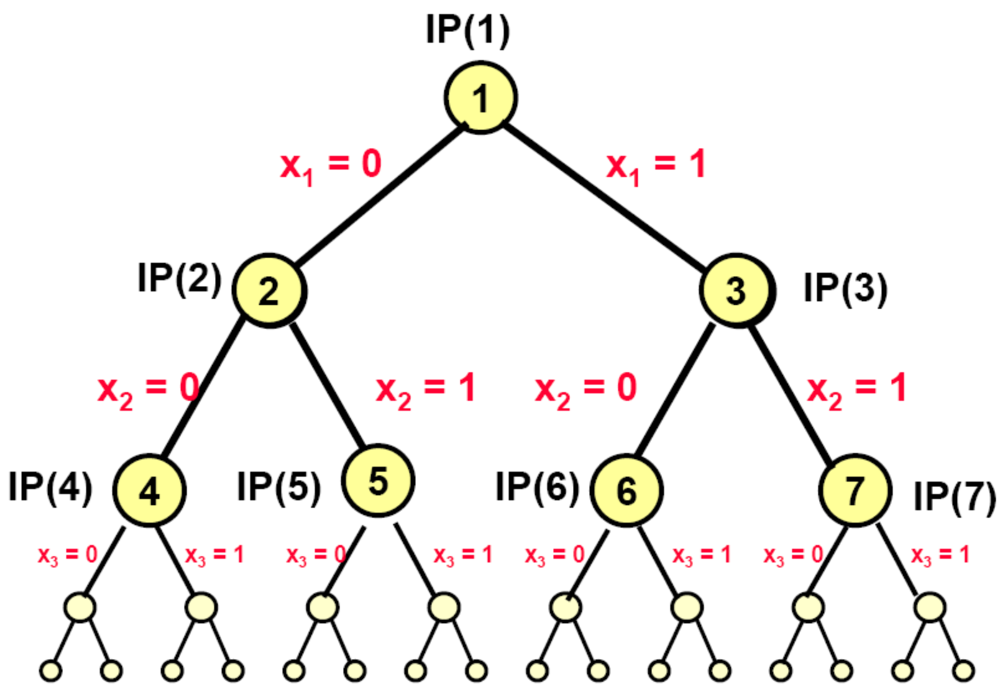
如何剪枝?

$$\max\ 24x_1 + 2x_2 + 20x_3 + 4x_4$$
$$\text{s.t. } 8x_1 + x_2 + 5x_3 + 4x_4 \le 9$$
$$x_i \in \{0,1\}, 1 \le i \le 4$$

- 以IP(2)为例，对应一个新的整数规划问题为

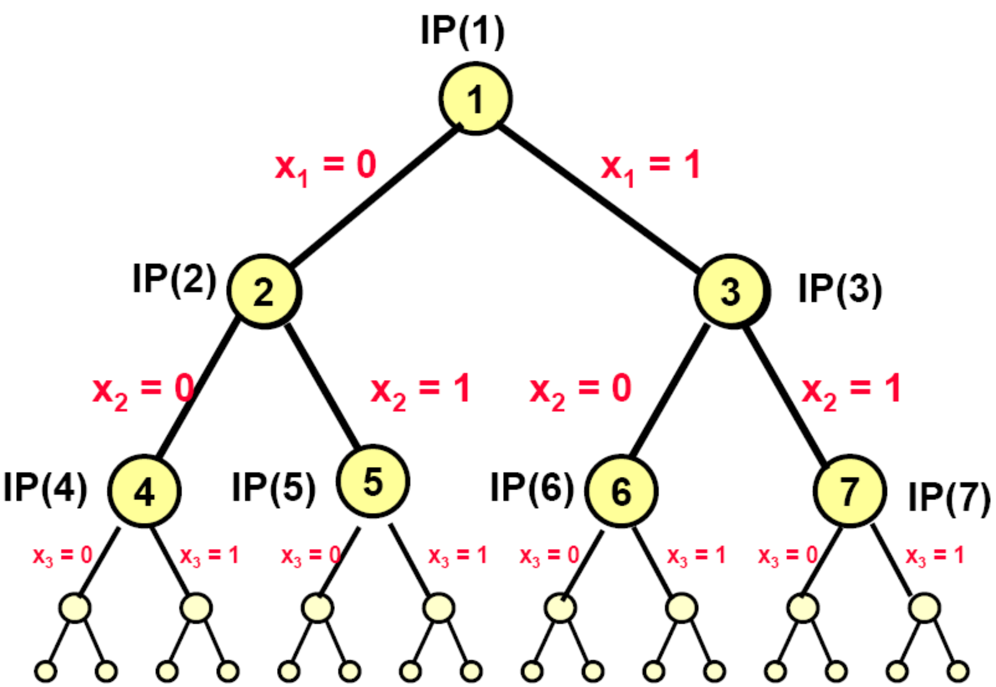
$$\max\ 24x_1 + 2x_2 + 20x_3 + 4x_4$$
$$\text{s.t. } 8x_1 + x_2 + 5x_3 + 4x_4 \le 9$$
$$x_1 = 0$$
$$x_i \in \{0,1\}, 2 \le i \le 4$$

- 如果某个节点可以不是最优解，它的子孙节点也不是



剪枝规则

- 子树可以被剪枝，如果以下条件之一成立
  - 已经求解到子树的根
  - 已经验证，子树根的整数解不是最优的
- 例如
  - 已知 IP(3) 的解为 26
  - 其子孙节点可以被剪枝
  - IP(4) 的解为24
  - IP(4) 的子孙节点也可以被剪枝



整数规划的线性规划松弛

- 如果决策变量的整数约束，IP 问题可以转化为一个 LP 问题，被称之为对应 IP 的 LP 松弛
- 背包问题的 LP 松弛可以使用贪心方法解决

$$\begin{aligned} \max\ & 24x_1 + 2x_2 + 20x_3 + 4x_4 \\ \text{s.t. } & 8x_1 + x_2 + 5x_3 + 4x_4 \le 9 \\ & x_i \in \{0,1\}, 1 \le i \le 4 \end{aligned}$$

$$\begin{aligned} \max\ & 24x_1 + 2x_2 + 20x_3 + 4x_4 \\ \text{s.t. } & 8x_1 + x_2 + 5x_3 + 4x_4 \le 9 \\ & 0 \le x_i \le 1, 1 \le i \le 4 \end{aligned}$$

item	1	2	3	4
value/lb.	\$3	\$2	\$4	\$1

IP与LP松弛解的关系

item	1	2	3	4
value/lb.	$3	$2	$4	$1

- 通过解 LP 松弛问题，可以确定对应 IP 的范围
  - $IP(k) \leq LP(k)$
  - 例如
    - ✓ $IP(1) < LP(1)$
    - ✓ $IP(4) = LP(4)$
  - 如果 $LP(k)$ 的最优解对 $IP(k)$ 是可行的，则 $IP(k) = LP(k)$
  - 如果 $LP(k)$ 的最优解对 $IP(k)$ 是不可行的，则 $IP(k) < LP(k)$

分支定界方法：Branch and Bound

```
while there is  some active nodes do
   select an active node j
   mark j as inactive
   Solve LP(j):  denote solution as x(j);
    Case 1 --   if z_LP(j) ≤ z_I then prune node j;
    Case 2 --   if z_LP(j) > z_I and
          if x(j) is feasible for IP(j)
          then Incumbent := x(j), and z_I := z_LP(j);
           then prune node j;
     Case 3 -- If  if z_LP(j) > z_I and
          if x(j) is not feasible for IP(j) then
          mark the children of node j as active
endwhile
```

- Active 顶点意味着需要进一步确定
- Z_I 表示到目前为止找到的最优整数解
- 因此，Z_I 可以帮助算法进行剪枝
  - $LP(j) \leq Z_I$
  - $LP(j) > Z_I$
    - $LP(j) = IP(j)$，更新 Z_I，并剪枝
    - 否则，不能被剪枝

B&B算法示例：IP(1)

$$\max\ 24x_1 + 2x_2 + 20x_3 + 4x_4$$
$$\text{s.t. } 8x_1 + x_2 + 5x_3 + 4x_4 \le 9$$
$$x_i \in \{0,1\}, 1 \le i \le 4$$

$$\downarrow$$

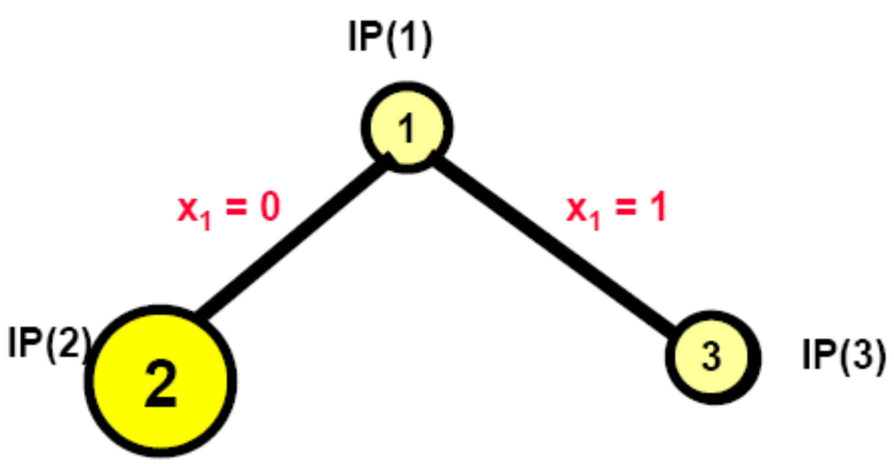
$$\max\ 24x_1 + 2x_2 + 20x_3 + 4x_4$$
$$\text{s.t. } 8x_1 + x_2 + 5x_3 + 4x_4 \le 9$$
$$0 \le x_i \le 1, 1 \le i \le 4$$

$$x_1 = \frac{1}{2}, x_2 = 0, x_3 = 1, x_4 = 0$$
$$Z_I = -\infty$$
$$Z_{LP(1)} = 32$$

B&B算法示例：IP(2)

$$\max\ 24x_1 + 2x_2 + 20x_3 + 4x_4$$
$$\text{s.t. } 8x_1 + x_2 + 5x_3 + 4x_4 \le 9$$
$$x_1 = 0$$
$$0 \le x_i \le 1, 2 \le i \le 4$$

$$Z_I = -\infty$$
$$Z_{LP(1)} = 32$$



求 LP(2) 的最优解

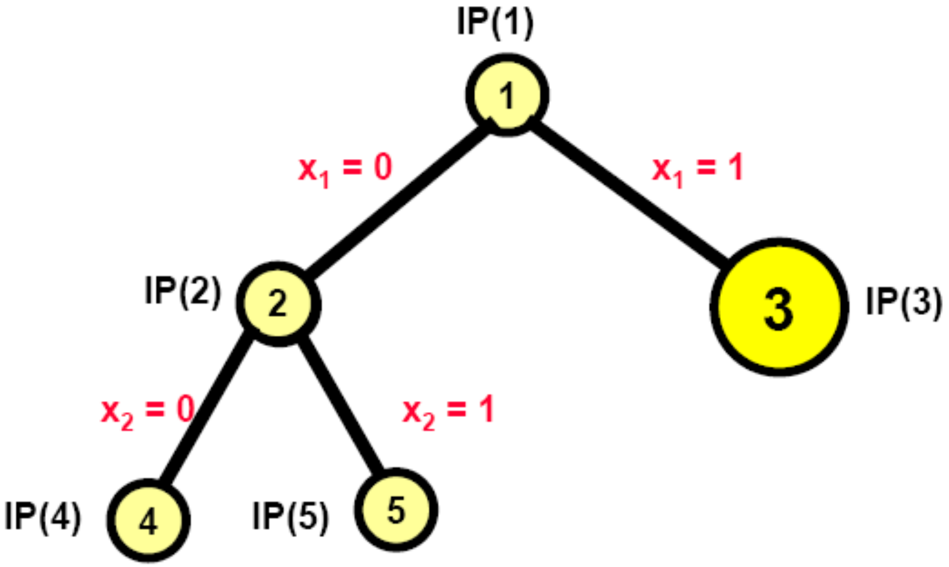
$$x_1 = 0, x_2 = 1, x_3 = 1, x_4 = \frac{3}{4}$$

因此, $Z_{LP(2)} = 25$

B&B算法示例：IP(3)

$$\max\ 24x_1 + 2x_2 + 20x_3 + 4x_4$$
$$\text{s.t. } 8x_1 + x_2 + 5x_3 + 4x_4 \le 9$$
$$x_1 = 1$$
$$0 \le x_i \le 1, 2 \le i \le 4$$

$$Z_I = -\infty$$



求 LP(3) 的最优解

$$x_1 = 1, x_2 = 0, x_3 = \frac{1}{5}, x_4 = 0$$

因此，$Z_{LP(3)} = 28$

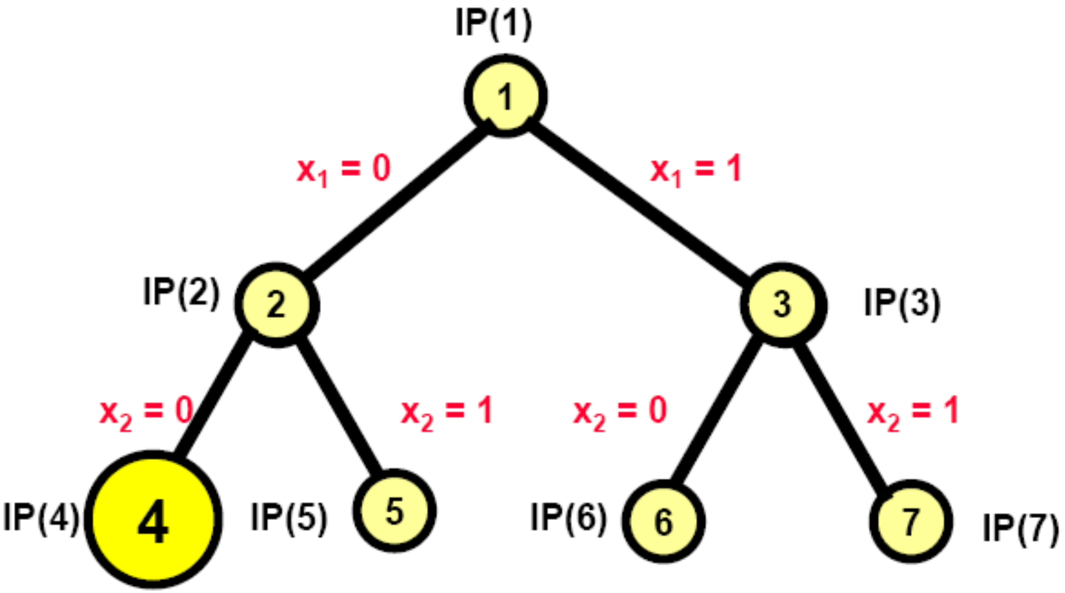
B&B算法示例：IP(4)

$$\begin{aligned} \max\ & 24x_1 + 2x_2 + 20x_3 + 4x_4 \\ \text{s.t.}\ & 8x_1 + x_2 + 5x_3 + 4x_4 \le 9 \\ & x_1 = 0, x_2 = 0 \\ & 0 \le x_i \le 1, 3 \le i \le 4 \end{aligned}$$

$Z_I = -\infty$

更新为

$Z_I = 24$



求 LP(4) 的最优解

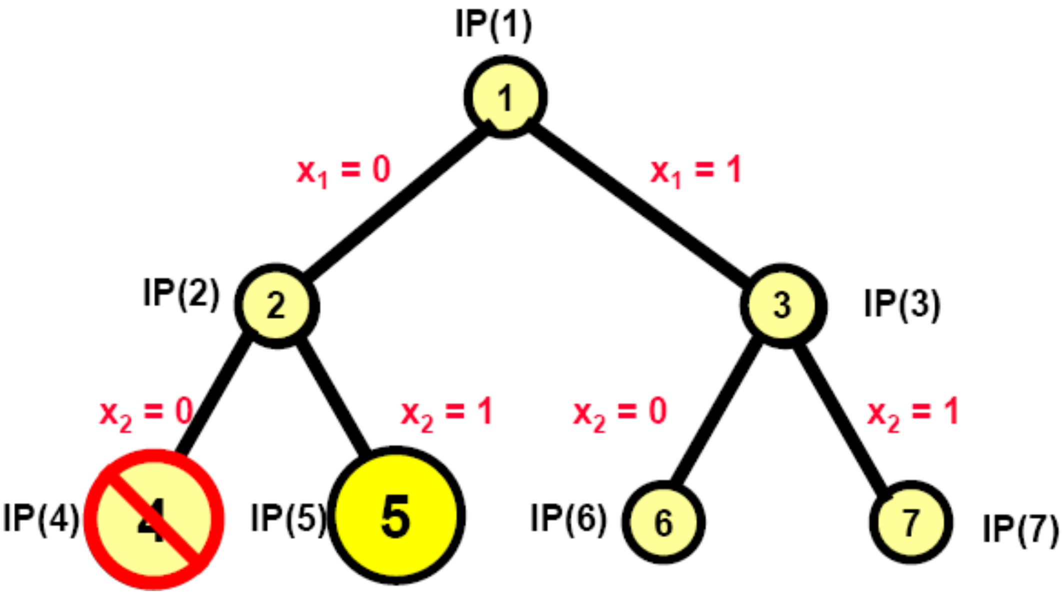
$x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 1$

因此，$Z_{IP(4)} = 24$

B&B算法示例：IP(5)

$$\begin{aligned} \max\ & 24x_1 + 2x_2 + 20x_3 + 4x_4 \\ \text{s.t.}\ & 8x_1 + x_2 + 5x_3 + 4x_4 \le 9 \\ & x_1 = 0, x_2 = 1 \\ & 0 \le x_i \le 1, 3 \le i \le 4 \end{aligned}$$

$Z_I = 24$



求 LP(5) 的最优解

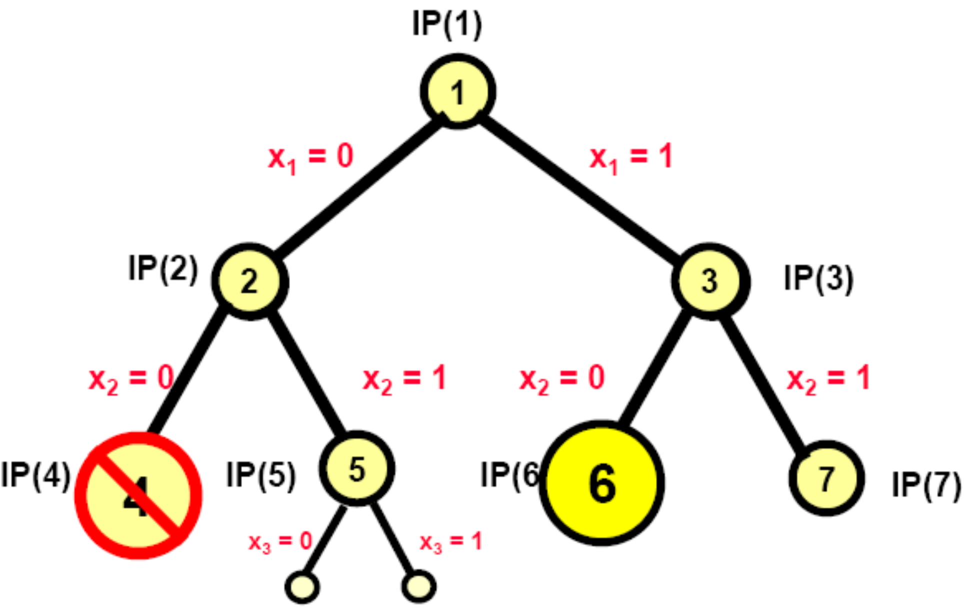
$$x_1 = 0, x_2 = 1, x_3 = 1, x_4 = \frac{3}{4}$$

因此，$Z_{LP(5)} = 25$

B&B算法示例：IP(6)

$$
\begin{aligned}
\max\ & 24x_1 + 2x_2 + 20x_3 + 4x_4 \\
\text{s.t. } & 8x_1 + x_2 + 5x_3 + 4x_4 \le 9 \\
& x_1 = 1, x_2 = 0 \\
& 0 \le x_i \le 1, 3 \le i \le 4
\end{aligned}
$$

$Z_I = 24$



求 LP(6) 的最优解

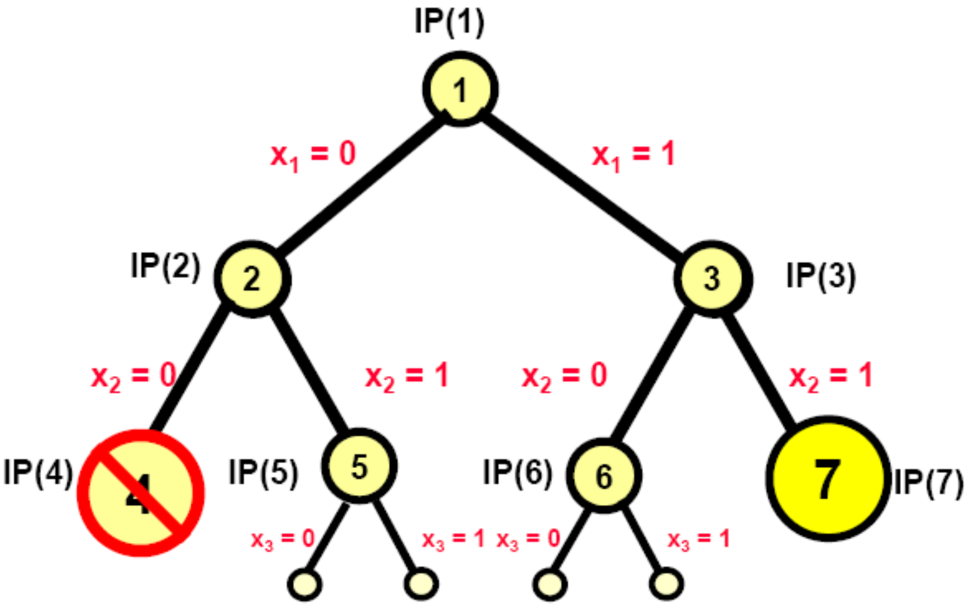
$$x_1 = 1, x_2 = 0, x_3 = \frac{1}{5}, x_4 = 0$$

因此，$Z_{LP(6)} = 28$

B&B算法示例：IP(7)

$$\max\ 24x_1 + 2x_2 + 20x_3 + 4x_4$$
$$\text{s.t. } 8x_1 + x_2 + 5x_3 + 4x_4 \le 9$$
$$x_1 = 1, x_2 = 1$$
$$0 \le x_i \le 1, 3 \le i \le 4$$

$Z_I = 24$
更新为
$Z_I = 26$



求 LP(7) 的最优解

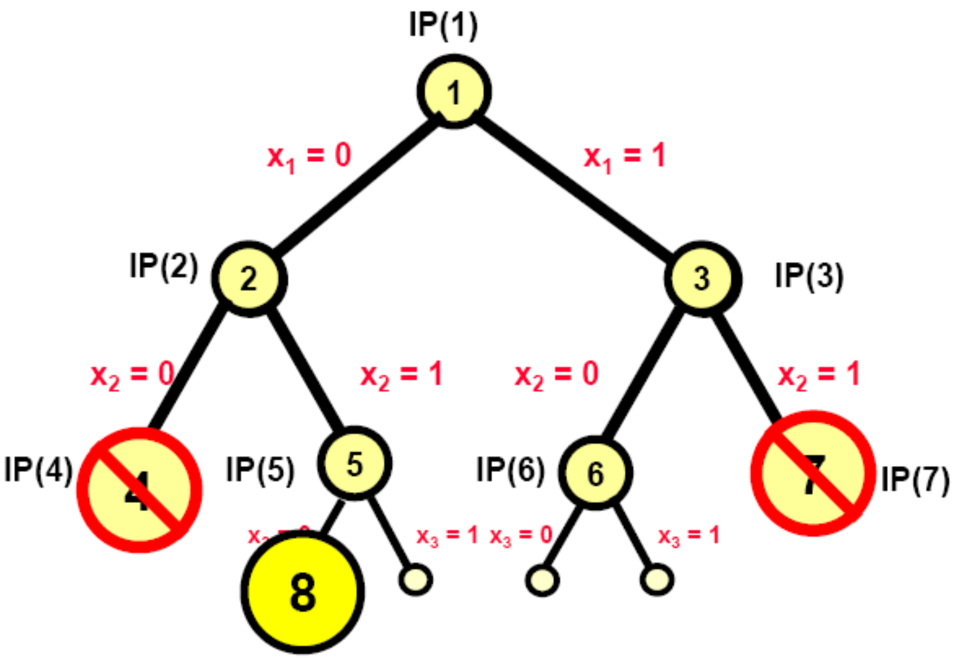
$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0$

因此，$Z_{LP(7)} = 26$

B&B算法示例：IP(8)

$$\max\ 24x_1 + 2x_2 + 20x_3 + 4x_4$$
$$\text{s.t. } 8x_1 + x_2 + 5x_3 + 4x_4 \le 9$$
$$x_1 = 0, x_2 = 1, x_3 = 0$$
$$0 \le x_4 \le 1$$

$Z_I = 26$



求 LP(8) 的最优解

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

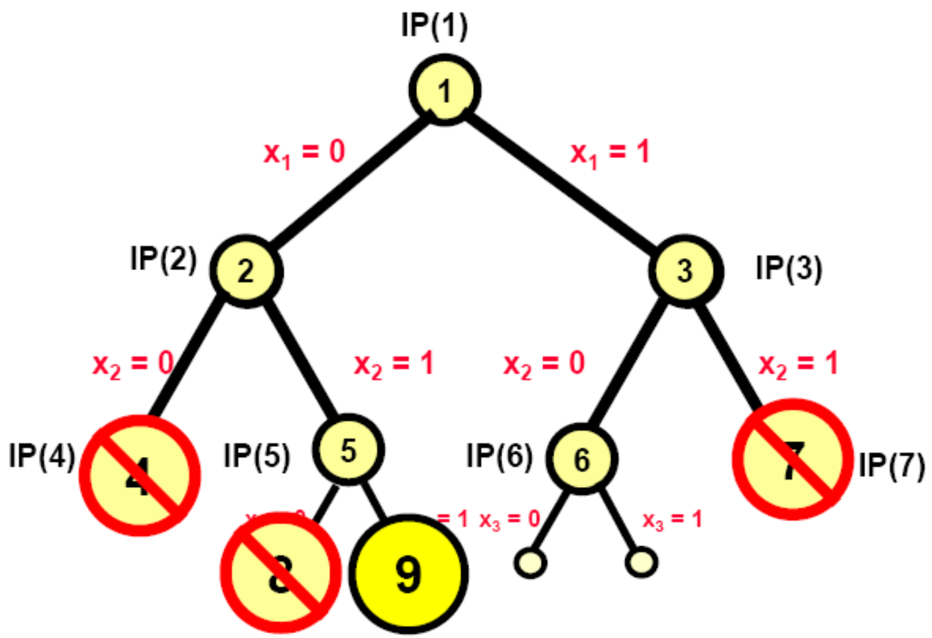
因此，$Z_{LP(8)} = 6$

$Z_{LP(8)} < Z_I$，Pruned

B&B算法示例：IP(9)

$$
\begin{aligned}
\max\ & 24x_1 + 2x_2 + 20x_3 + 4x_4 \\
\text{s.t.}\ & 8x_1 + x_2 + 5x_3 + 4x_4 \le 9 \\
& x_1 = 0, x_2 = 1, x_3 = 1 \\
& 0 \le x_4 \le 1
\end{aligned}
$$

$$Z_I = 26$$



求 LP(9) 的最优解

$$x_1 = 0, x_2 = 1, x_3 = 1, x_4 = \frac{3}{4}$$

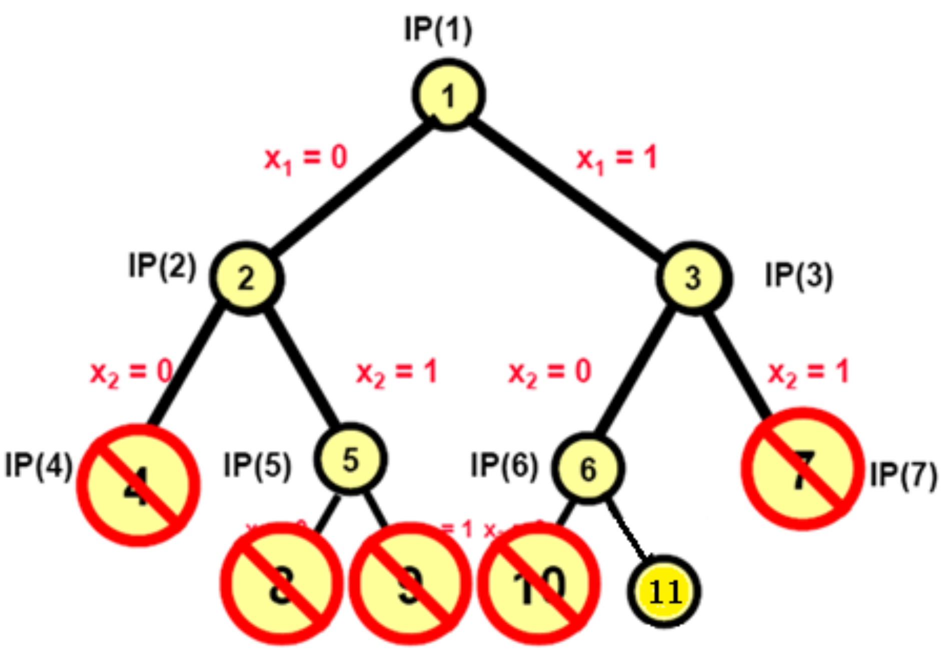
因此，$Z_{LP(9)} = 25$

$Z_{LP(9)} < Z_I$，Pruned

B&B算法示例：IP(10)

$$\max\ 24x_1 + 2x_2 + 20x_3 + 4x_4$$
$$\text{s.t. } 8x_1 + x_2 + 5x_3 + 4x_4 \le 9$$
$$x_1 = 1, x_2 = 0, x_3 = 0$$
$$0 \le x_4 \le 1$$

$$Z_I = 26$$



求 LP(10) 的最优解

$$x_1 = 1, x_2 = 0, x_3 = 0, x_4 = \frac{1}{4}$$

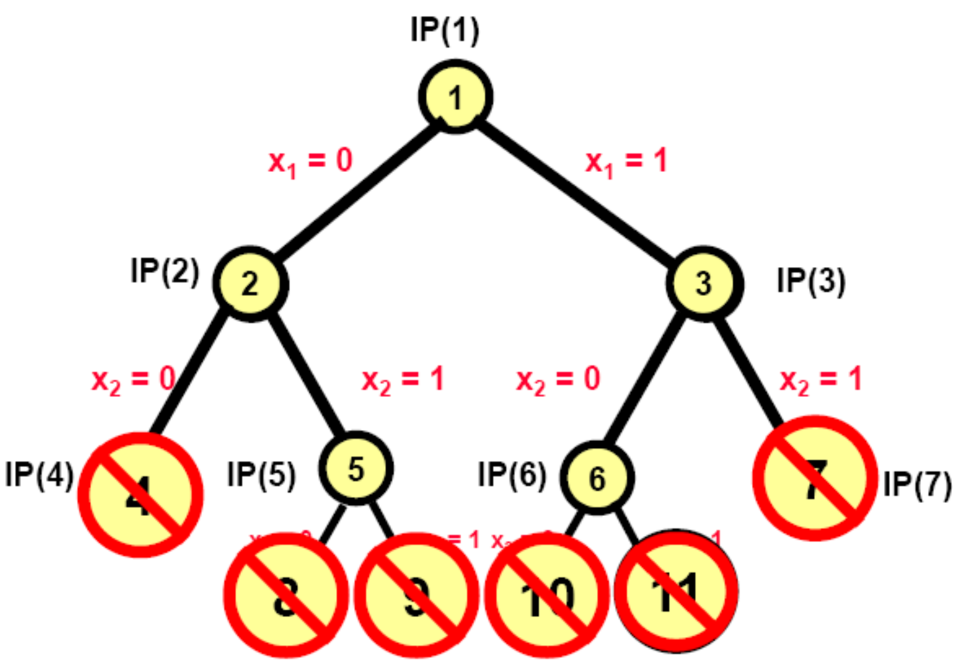
因此，$Z_{LP(10)} = 25$

$Z_{LP(10)} < Z_I$，Pruned

B&B算法示例：IP(11)

$$\begin{aligned} \max\ & 24x_1 + 2x_2 + 20x_3 + 4x_4 \\ \text{s.t.}\ & 8x_1 + x_2 + 5x_3 + 4x_4 \le 9 \\ & x_1 = 1, x_2 = 0, x_3 = 1 \\ & 0 \le x_4 \le 1 \end{aligned}$$

$Z_I = 26$



求 LP(11) 的最优解无解，Pruned

B&B 算法总结

- B&B 算法加快了 IP 问题求解的速度
  - 在 31 个节点的枚举树中，仅需要评估 11 个节点（LP）
  - 节点数量越多，剪枝效率可能越高
  - 然而剪枝效率的影响因素众多，比如问题规模、节点访问顺序
- 如何提升剪枝效率?
  - 通过前面的例子，如果能够尽快找到最优解 26，或者接近于最优解，那么剪枝效率会更高
  - 启发式地智能选择合适的分支，尽快找到一个比较大的上界
  - 另一种技术是“取整”
    - $x_1 + x_2 \leq 1.9 \Rightarrow x_1 + x_2 \leq 1$
    - $Z_{IP} \leq Z_{LP} = 5.5 \Rightarrow Z_{IP} \leq Z_{LP} = 5$

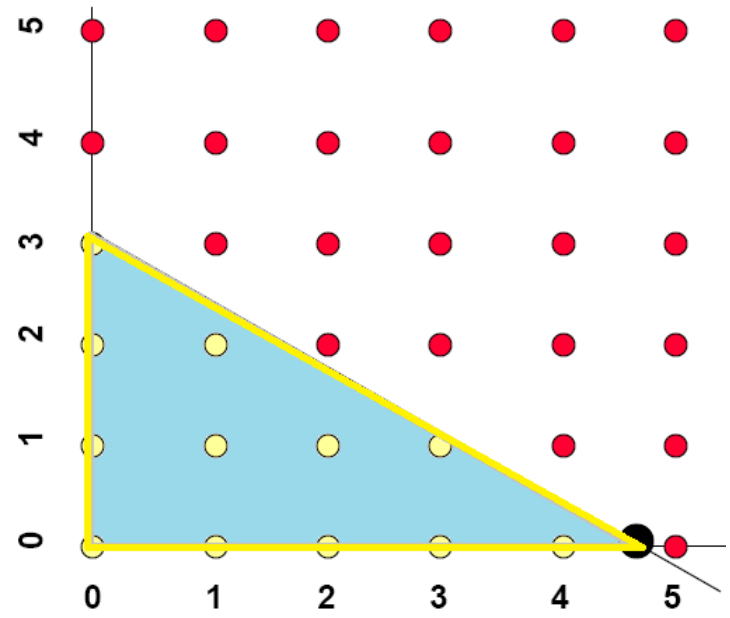
课程提纲

Content

有效不等式

- IP 的有效不等式是不减少任何可行整数解的约束
- 例如

$$\max\ z = 3x + 4y$$
$$\text{s.t. } 5x + 8y \le 24$$
$$0 \le x, y \in \mathbb{Z}$$



  - $x \le 5$ 是有效不等式吗?
  - $x \le 4$ 是有效不等式吗?

- 一个有效不等式又被称之为割平面或者割

四舍五入

- 整数变量的小数界可以被截断
  - $x \le 1.9 \Rightarrow x \le 1$
- 给定一个约束，它涉及的整系数有非1公约数
  - $3x + 6y + 9z \le 11 \Rightarrow x + 2y + 3z \le \lfloor \frac{11}{3} \rfloor \Rightarrow x + 2y + 3z \le 3$
- 对涉及非负系数的约束
  - $\sum_i a_i x_i \le b \Rightarrow \sum_i \lfloor \frac{a_i}{c} \rfloor x_i \le \sum_i \frac{a_i}{c} x_i \le \frac{b}{c}$
  - 左边是整系数的，右边也可以进行四舍五入
  - 找到新的有效不等式，并不意味着原来的约束就要被删除
- 四舍五入是发现有效不等式的有效途径之一

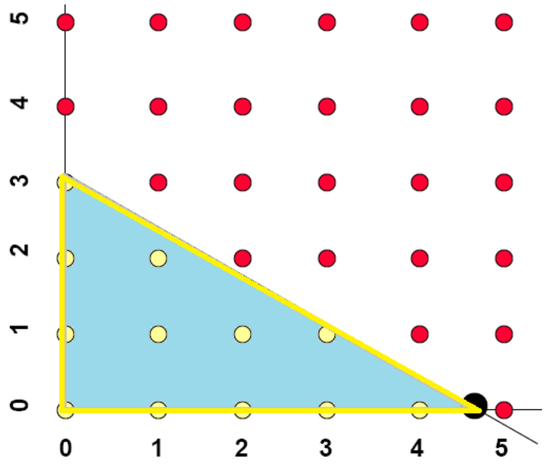
Gomory割

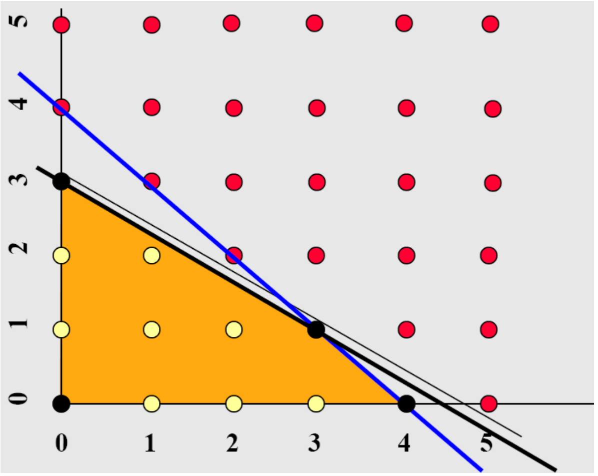
- Case 1：所有系数在 0 到 1 之间
  - $0.2x_1 + 0.3x_2 + 0.3x_3 + 0.5x_4 + x_5 = 1.8$
  - $0.2x_1 + 0.3x_2 + 0.3x_3 + 0.5x_4 \geq 0.8$
- Case 2：所有系数均为非负
  - $1.2x_1 + 0.3x_2 + 2.3x_3 + 2.5x_4 + x_5 = 4.8$
  - $0.2x_1 + 0.3x_2 + 0.3x_3 + 0.5x_4 \geq 0.8$
- Case 3: 一般情况
  - $1.2x_1 - 1.3x_2 - 2.4x_3 + 11.8x_4 + x_5 = 2.9$
  - $x_1 - 2x_2 - 3x_3 + 11x_4 + x_5 \leq 2$
  - $0.2x_1 + 0.7x_2 + 0.6x_3 + 0.8x_4 \geq 0.9$
- Gomory 割是另一种寻找有效不平等的途径

凸包

- 凸包是包含所有整数解的最小松弛 LP 的可行域

$$\begin{aligned} \max\ z &= 3x + 4y \\ \text{s.t. } 5x + 8y &\le 24 \\ 0 \le x, y &\in \mathbb{Z} \end{aligned}$$

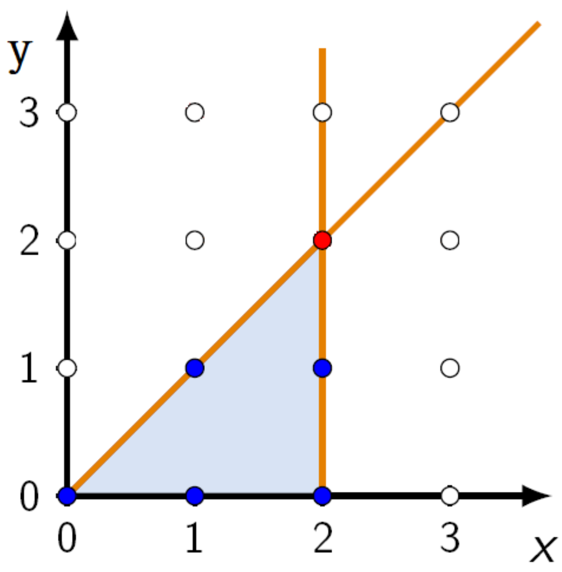


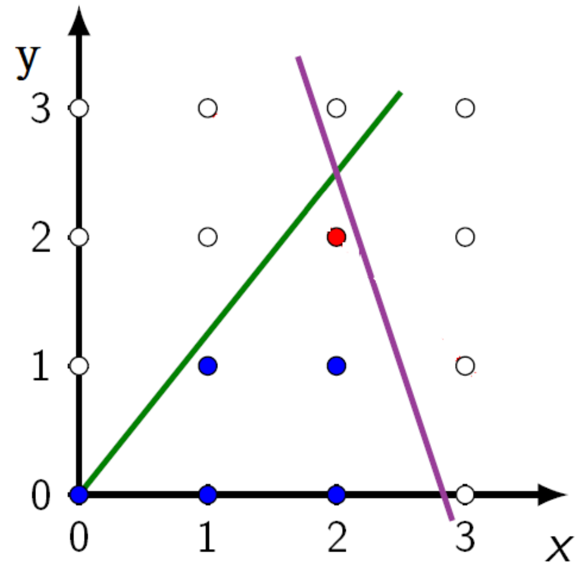


$$\begin{aligned} \max\ z &= 3x + 4y \\ \text{s.t. } 5x + 8y &\le 24 \\ x + y &\le 4 \\ 2x + 3y &\le 9 \\ 0 \le x, y &\in \mathbb{Z} \end{aligned}$$

凸包示例

$$\begin{aligned} \max\ & z = x + y \\ \text{s.t. } & -5x + 4y \le 0 \\ & 6x + 2y \le 17 \\ & 0 \le x, y \in \mathbb{Z} \end{aligned}$$





$$\begin{aligned} \max\ & z = x + y \\ \text{s.t. } & -5x + 4y \le 0 \\ & 6x + 2y \le 17 \\ & x \le 2 \\ & y \le x \\ & 0 \le x, y \in \mathbb{Z} \end{aligned}$$

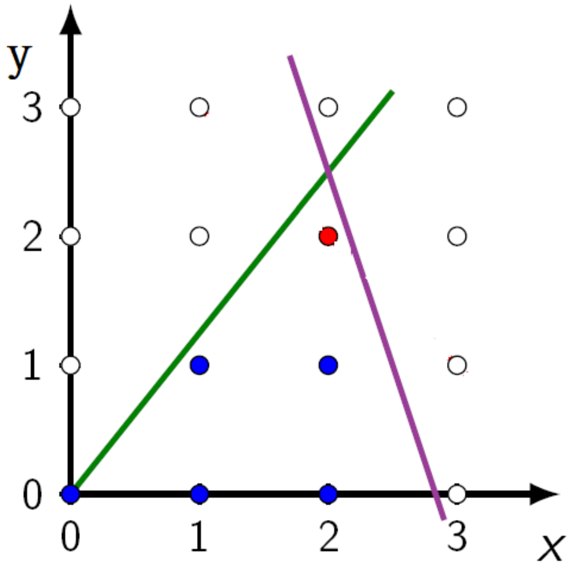
凸包的作用

- 定义在凸包上 LP 的解和 IP 的解是相同的
- 试图找到凸包几乎是不可能的
  - 可能有太多的约束
  - 而且找全这些约束很难
- 找到凸包中有用的约束
  - 保证 LP 的解和 IP 的解是相同的
  - 但是做到这一点和发现凸包一样，是很困难的
- 找到有用的有效不等式
  - 不断消减 LP 问题的可行域，使得每次 LP 的最优解被剔除
  - 使得 LP 的解越来越接近于 IP 的解
  - 这是常用并且可行的方法

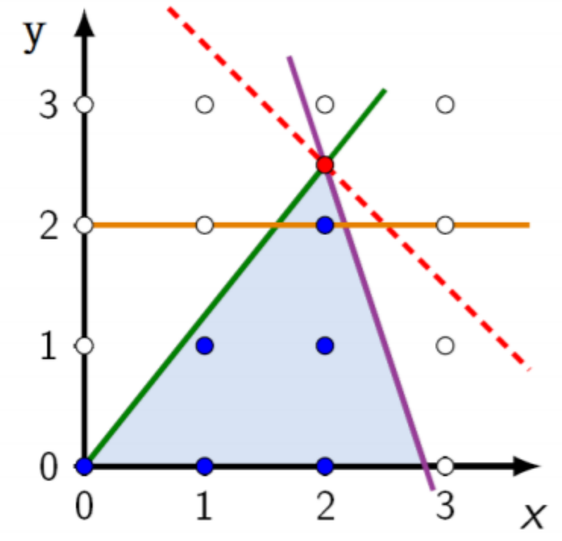
割平面法

- 割平面法
  - Step 1：解松弛的 LP（单纯形法或者对偶单纯形法）
  - Step 2：如果其解为整数解，问题解决
  - Step 3：如果不是整数解，则找一个能把它剔除掉的有效不等式
  - Step 4：获得新的 LP 问题，返回Step 1

$$
\begin{aligned}
\max\ & z = x + y \\
\text{s.t.}\ & -5x + 4y \le 0 \\
& 6x + 2y \le 17 \\
& 0 \le x, y \in \mathbb{Z}
\end{aligned}
$$



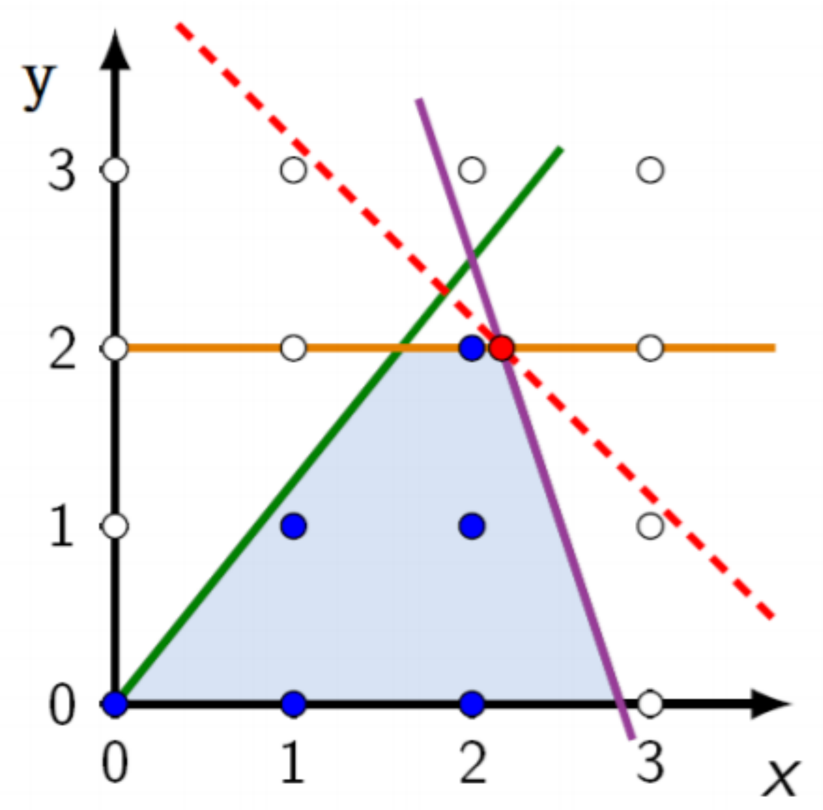
示例



$$
\begin{aligned}
\max\ & z = x + y \\
\text{s.t.}\ & -5x + 4y \le 0 \\
& 6x + 2y \le 17 \\
& y \le 2 \\
& 0 \le x, y \in \mathbb{Z}
\end{aligned}
$$

- $y \le 2$ 是一个有效不等式，将松弛 LP 的非整数解剔除了，但是又没有影响到 IP 问题的解
    - 实际中，找有效不等式需基于松弛 LP 的解推断出一个有效不等式
    - 可以参考运筹学相关内容
- 运用单纯形法或者对偶单纯形法继续求解松弛的 LP

示例（续）

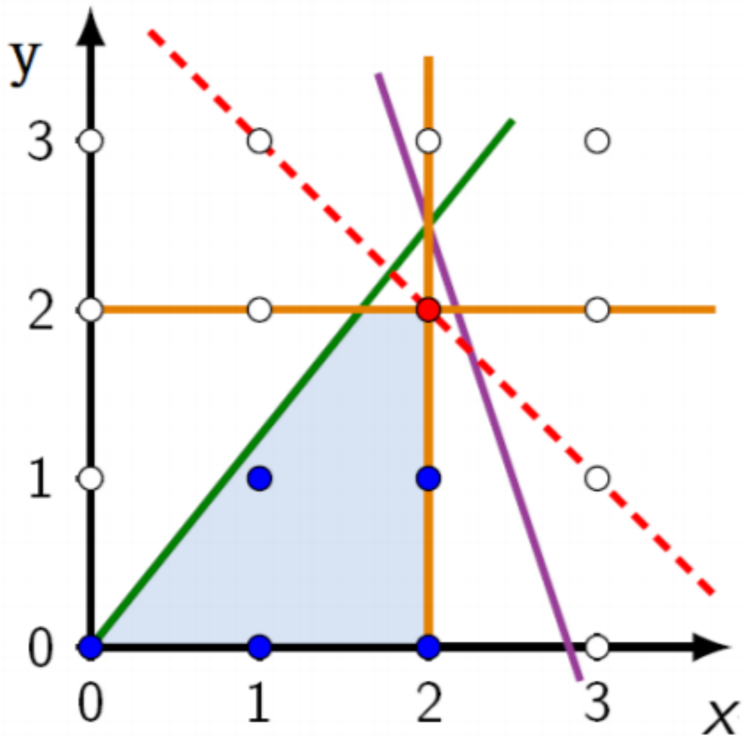


$$
\begin{aligned}
\max\ & z = x + y \\
\text{s.t.}\ & -5x + 4y \le 0 \\
& 6x + 2y \le 17 \\
& y \le 2 \\
& 0 \le x, y \in \mathbb{Z}
\end{aligned}
$$

最优解为 4.16667

- 此时，松弛 LP 的解还不是整数解
- 寻找新的有效不等式

示例（续）



$$\max z = x + y$$

$$\text{s.t.} \ -5x + 4y \leq 0$$

$$6x + 2y \leq 17$$

$$y \leq 2$$

$$x \leq 2$$

$$0 \leq x, y \in \mathbb{Z}$$

最优解为 4

- 松弛 LP 的解是整数解：$x = 2, y = 2$
- 即为原 IP 问题的最优解

本章小结

- 整数规划
  - 整数规划是一类离散优化问题，又称为组合优化
  - 具有普遍的应用，如 TSP、SCP 和 Fiver 等等
  - 类型也很丰富
- 求解整数规划问题
  - 分支限界方法
  - 割平面法
- 整数规划问题很多是 NP–Hard 问题
  - 近似方法求解
  - 整数规划求解效率和约束相关，因此合理描述约束非常重要
  - 可以选用一些开源工具或者软件求解，比 Lingo 和 Lindo