5.4 分支限界

组合优化问题的相关概念

目标函数（极大化或极小化）

约束条件

搜索空间中满足约束条件的解称为可行解

使得目标函数达到极大(或极小)的解称为最优解

5.4. 1 背包问题

$$\max \quad x_1 + 3x_2 + 5x_3 + 9x_4$$

$$2x_1 + 3x_2 + 4x_3 + 7x_4 \le 10$$

$$x_i \in N, i = 1,2,3,4$$

北京大學

分支限界技术(极大化)

设立代价函数

函数值以该结点为根的搜索树中的所有可行解的目标函数值的上界

父结点的代价不小于子结点的代价

设立界

代表当时已经得到的可行解的目标函数的最大值

界的设定初值可以设为 0

可行解的目标函数值大于当时的界，进行更新

搜索中停止分支的依据

不满足约束条件或者其代价函数小于当时的界

实例：背包问题

背包问题的实例：

$$\max x_1 + 3x_2 + 5x_3 + 9x_4$$

$$2x_1 + 3x_2 + 4x_3 + 7x_4 \le 10$$

$$x_i \in \mathrm{N}, i = 1,2,3,4$$

对变元重新排序使得

$$\frac{v_i}{w_i} \ge \frac{v_{i+1}}{w_{i+1}}$$

排序后实例 $\max 9x_1 + 5x_2 + 3x_3 + x_4$

$$7x_1 + 4x_2 + 3x_3 + 2x_4 \le 10$$

$$x_i \in \mathrm{N}, i = 1,2,3,4$$

代价函数与分支策略确定

结点$<x_1, x_2, \ldots, x_k>$ 的代价函数

$$\sum_{i=1}^{k} v_i x_i + \left(b - \sum_{i=1}^{k} w_i x_i\right)\frac{v_{k+1}}{w_{k+1}} \qquad \text{若对某个} j > k \text{有} b - \sum_{i=1}^{k} w_i x_i \geq w_j$$

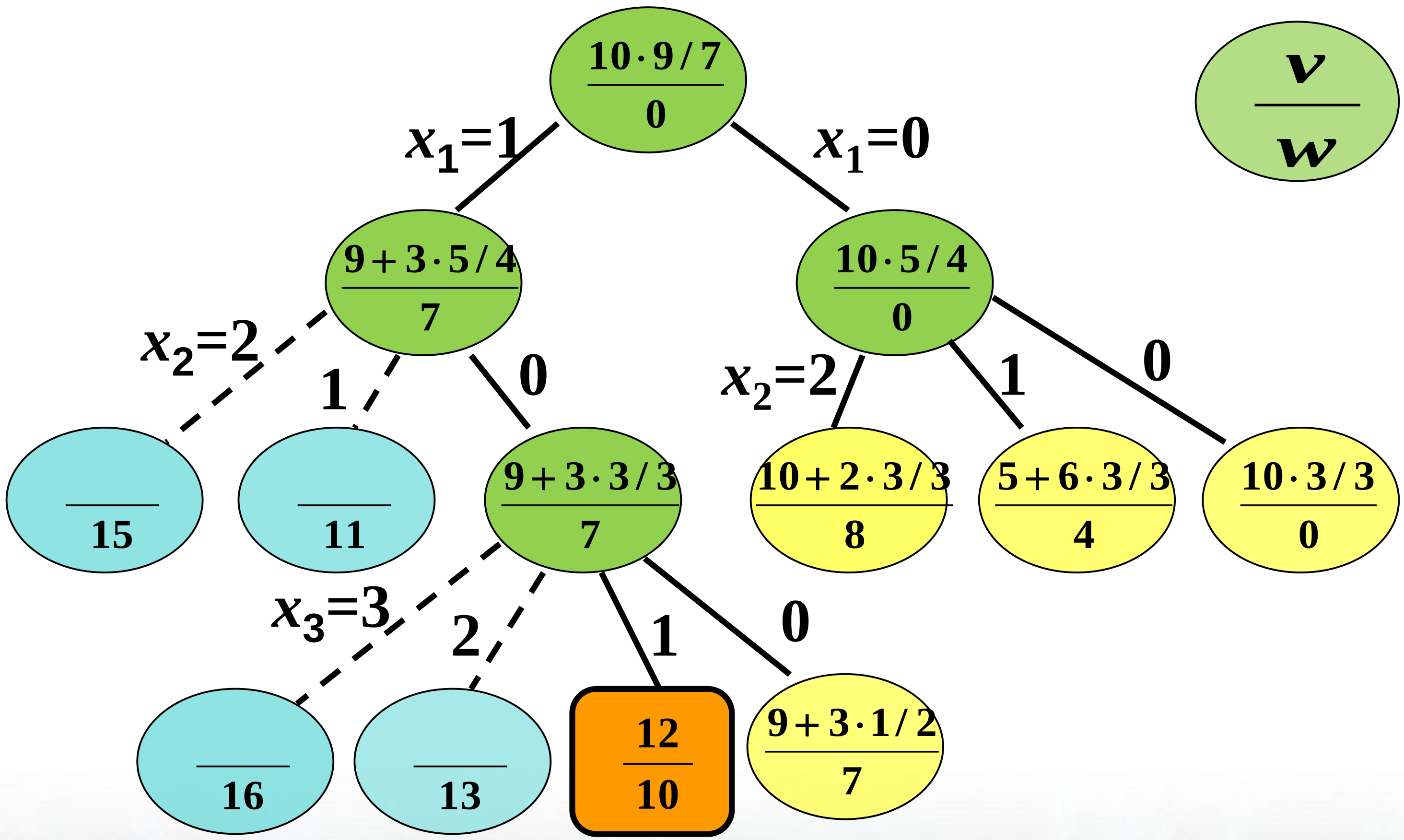
$$\sum_{i=1}^{k} v_i x_i \qquad \text{否则}$$

分支策略----深度优先

$$\max\ 9x_1 + 5x_2 + 3x_3 + x_4$$

$$7x_1 + 4x_2 + 3x_3 + 2x_4 \le 10,\quad x_i \in \mathrm{N}, i = 1,2,3,4$$

实例



5.4.2 最大团问题

问题：给定无向图G=<V, E>, 求G中的最大团.

相关知识：

无向图G = <V, E>,

G的子图：G'= <V',E'>, 其中$V' \subseteq V$, $E' \subseteq E$,

G的补图：$\breve{G}$ =<V, E'>, E'是E关于完全图边集的补集

G中的团：G 的完全子图

G 的点独立集：G 的顶点子集A，且$\forall u,v \in A$, $(u,v) \notin E$.

最大团：顶点数最多的团

最大点独立集：顶点数最多的点独立集

命题：U是G 的最大团当且仅当U是$\breve{G}$ 的最大点独立集

算法设计

结点$<x_1, x_2, \ldots, x_k>$的含义：

已检索 k 个顶点，其中 $x_i=1$ 对应的顶点在当前的团内

搜索树为子集树

约束条件：该顶点与当前团内每个顶点都有边相连

界：当前图中已检索到的极大团的顶点数

代价函数：目前的团扩张为极大团的顶点数上界

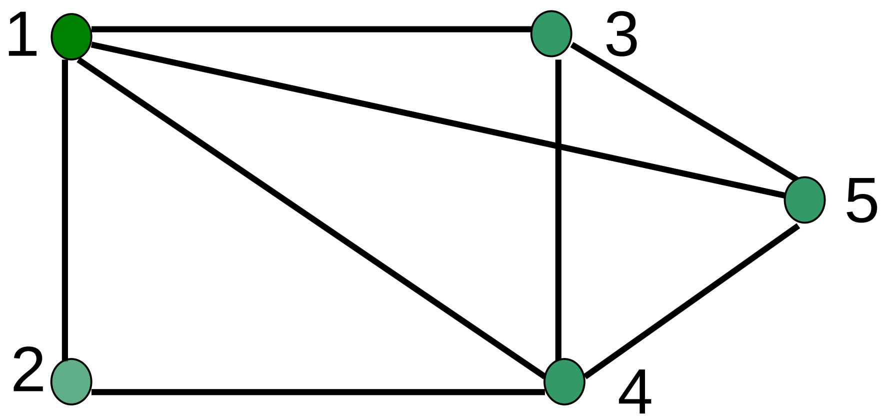
$$F = C_n + n - k$$

其中 C_n 为目前团的顶点数（初始为0），

k 为结点层数

时间：$O(n2^n)$

最大团的实例



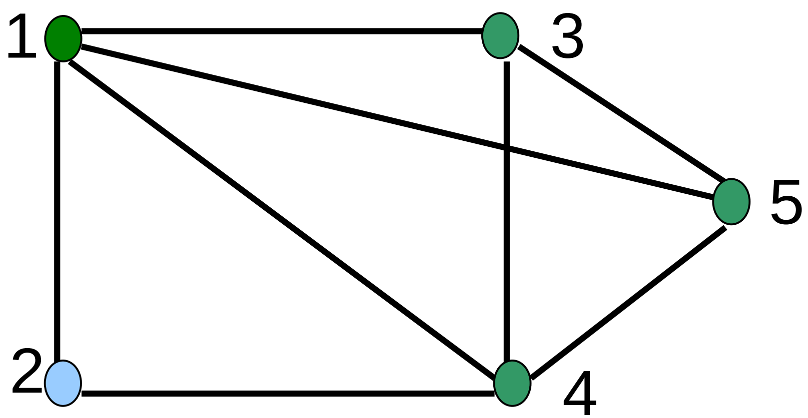
顶点编号顺序为 **1, 2, 3, 4, 5,**

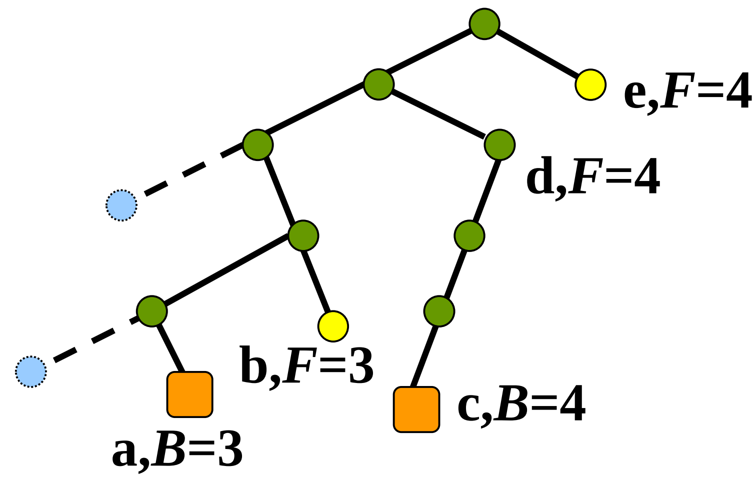
对应 x_1, x_2, x_3, x_4, x_5, x_i=**1** 当且仅当 i 在团内

分支规定左子树为**1**，右子树为**0.**

B 为界，F 为代价函数值.

实例求解





a： 得第一个极大团 **{ 1, 2, 4 }**, 顶点数为**3**, 界为**3**;

b： 代价函数值 ***F*** **= 3**, 回溯;

c： 得第二个极大团**{1, 3, 4, 5}**, 顶点数为**4**, 修改界为**4**;

d： 不必搜索其它分支, 因为***F*** **= 4**, 不超过界;

e： ***F*** **= 4**, 不必搜索.

最大团为 **{1, 3, 4, 5}**, 顶点数为 **4.**

5.4.3 货郎问题

问题：给定n个城市集合$C=\{c_1,c_2,\ldots,c_n\}$, 从一个城市到另一个城市的距离 d_{ij} 为正整数，求一条最短且每个城市恰好经过一次的巡回路线.

货郎问题的类型：有向图、无向图.

设巡回路线从1开始，

解向量为$<i_1, i_2, \ldots , i_{n-1}>$,

其中 $i_1, i_2, \ldots , i_{n-1}$为$\{ 2, 3, \ldots , n \}$的排列.

搜索空间为排列树，结点$<i_1, i_2, \ldots , i_k>$ 表示得到 k 步路线

北京大学

算法设计

约束条件：令$B = \{ i_1, i_2, \ldots , i_k \}$, 则

$$i_{k+1} \in \{ 2, \ldots , n \} - B$$

界：当前得到的最短巡回路线长度

代价函数：设顶点 c_i 出发的最短边长度为 l_i，d_j 为选定巡回路线中第 j 段的长度，则

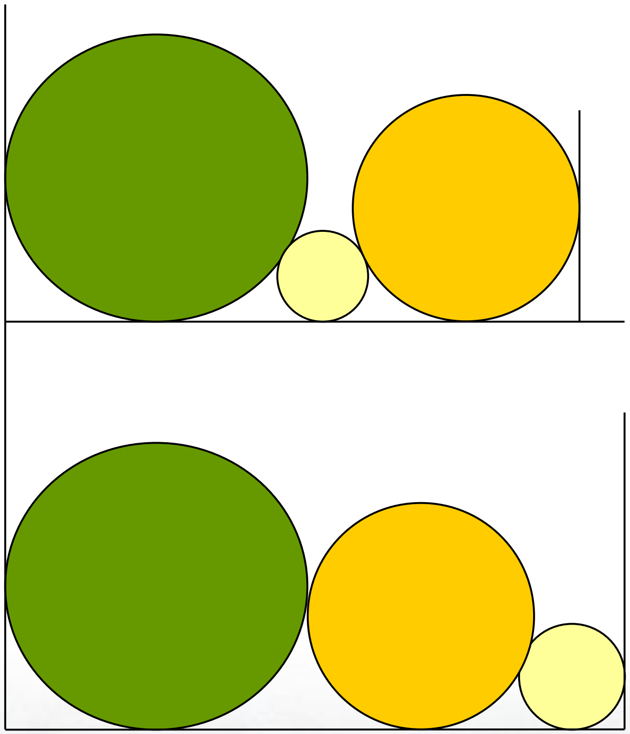
$$L = \sum_{j=1}^{k} d_j + l_{i_k} + \sum_{i_j \notin B} l_{i_j}$$

为部分巡回路线扩张成全程巡回路线的长度下界

时间 $O(n!)$：计算$O((n-1)!)$次，代价函数计算$O(n)$

5.4.4 圆排列问题

问题：给定n个圆的半径序列, 将各圆与矩形底边相切排列, 求具有最小长度 l_n 的圆的排列顺序.



解为<i_1, i_2, … , i_n>为1, 2, … , n的排列，解空间为排列树.

部分解向量 <i_1, i_2, … , i_k>：表示前 k 个圆已排好. 令B={ i_1, i_2, … , i_k },下一个圆选择 i_{k+1}.

约束条件：$i_{k+1} \in \{1, 2, \ldots, n\} - B$

界：当前得到的最小圆排列长度

代价函数符号说明

k：算法完成第 k 步，已经选择了第1— k 个圆

r_k：第 k 个圆的半径

d_k：第 k–1 个圆到第 k 个圆的圆心水平距离，k>1

x_k：第 k 个圆的圆心坐标，规定 x_1=0,

l_k：第 1— k 个圆的排列长度

L_k：放好 1—k 个圆以后，对应结点的代价函数值

$L_k \leq l_n$

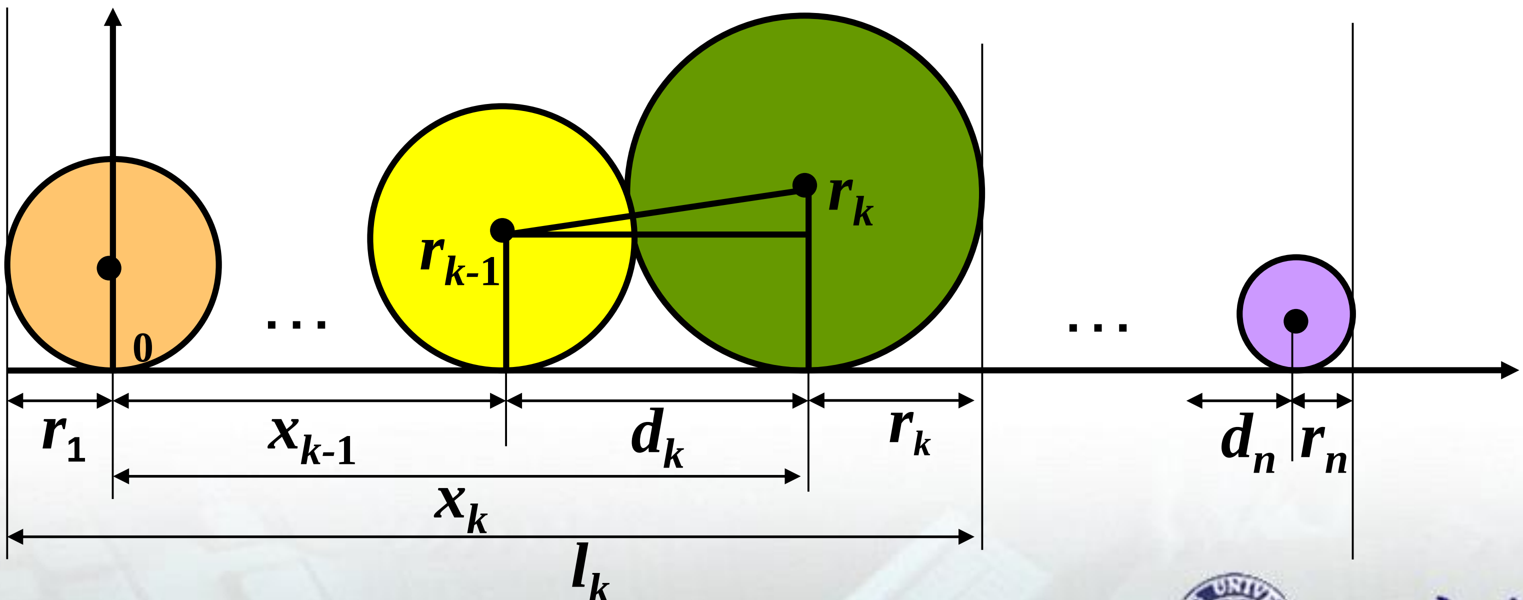
有关量的计算

$$d_k = \sqrt{(r_{k-1}+r_k)^2-(r_{k-1}-r_k)^2} = 2\sqrt{r_{k-1}r_k}$$

$$x_k = x_{k-1}+d_k, \qquad l_k = x_k + r_k + r_1$$

$$l_n = x_k + d_{k+1} + d_{k+2} + \ldots + d_n + r_n + r_1$$

$$= x_k + 2\sqrt{r_k r_{k+1}} + 2\sqrt{r_{k+1}r_{k+2}} + \ldots + 2\sqrt{r_{n-1}r_n} + r_n + r_1$$



代价函数

排列长度是l_n，L是代价函数：

$$
\begin{aligned}
l_n &= x_k + 2\sqrt{r_k r_{k+1}} + 2\sqrt{r_{k+1} r_{k+2}} + \ldots + 2\sqrt{r_{n-1} r_n} + r_n + r_1 \\
&\geq x_k + 2(n-k)r + r + r_1
\end{aligned}
$$

$$L = x_k + (2n - 2k + 1)r + r_1$$

$$r = \min(r_{i_j}, r_k) \quad i_j \in \{1, 2, \ldots, n\} - B$$

$$B = \{\, i_1, i_2, \ldots, i_k \,\},$$

时间：$O(n\ n!)=O((n+1)!)$

实例：计算过程

$R = \{1, 1, 2, 2, 3, 5\}$

取排列 <1, 2, 3, 4, 5, 6>,

半径排列为：1, 1, 2, 2, 3, 5，结果见下表和下图

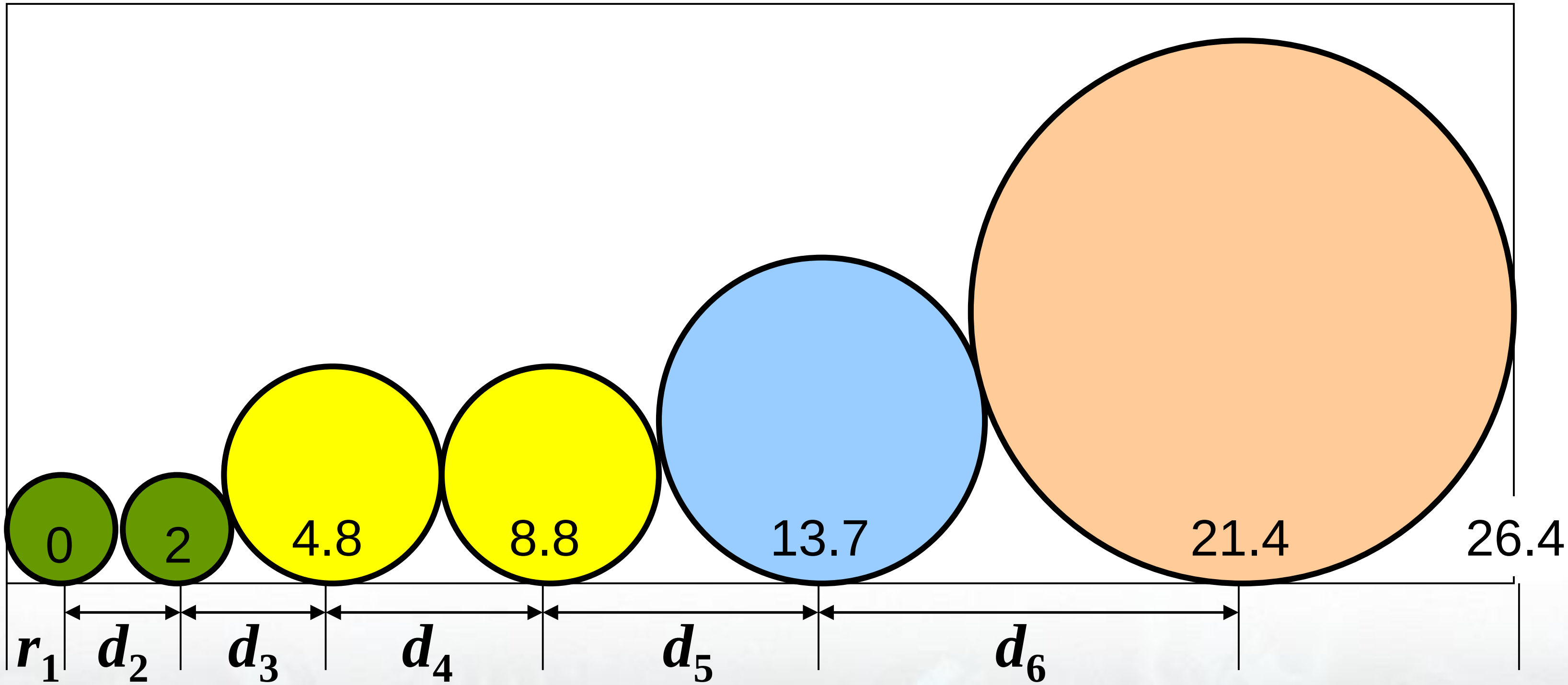
k	r_k	d_k	x_k	l_k	L_k
1	1	0	0	2	12
2	1	2	2	4	12
3	2	2.8	4.8	7.8	19.8
4	2	4	8.8	11.8	19.8
5	3	4.9	13.7	17.7	23.7
6	5	7.7	21.4	27.4	27.4

实例：图示

R = {1, 1, 2, 2, 3, 5 }

取排列 <1, 2, 3, 4, 5, 6>, 半径排列为：1, 1, 2, 2, 3, 5,

最短长度 $l_6 = 27.4$



5.4.5 连续邮资问题

问题：给定n种不同面值的邮票，每个信封至多m张， 试给出邮票的最佳设计，使得从1开始，增量为1的连续邮资区间达到最大？

实例：n=5，m=4，
面值 X_1=<1,3,11,15,32>，邮资连续区间为{ 1, 2, …,70 }
面值 X_2=<1,6,10,20,30>，邮资连续区间为{1, 2, 3, 4}

可行解：$<x_1, x_2, \ldots, x_n>$，$x_1=1$， $x_1< x_2< \ldots < x_n$
约束条件：在结点$<x_1, x_2, \ldots, x_i>$处，邮资最大连续区间为{ 1, … , r_i }，x_{i+1}的取值范围是{ x_i+1, … , r_i+1}

r_i 的计算

$y_i(j)$：用至多 m 张面值 x_i 的邮票加上 $x_1, x_2, \ldots, x_{i-1}$ 面值的邮票贴 j 邮资时的最少邮票数，则

$$y_i(j) = \min_{1 \le t \le m} \{t + y_{i-1}(j - t\,x_i)\}$$

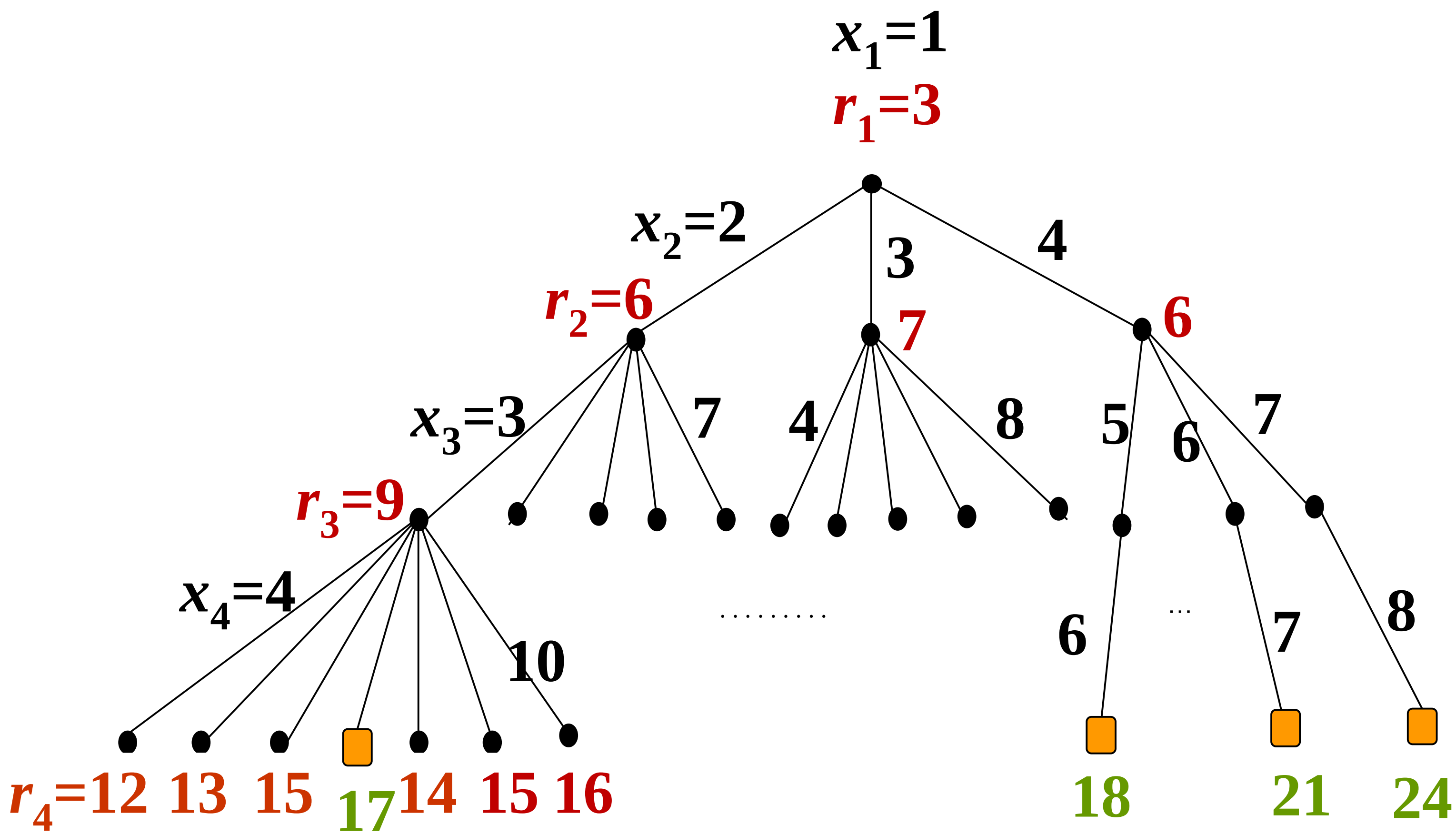
$$y_1(j) = j$$

$$r_i = \min\{ j \mid y_i(j) \le m, y_i(j+1) > m\}$$

搜索策略：深度优先

界：*max*，*m* 张邮票可付的连续区间的最大邮资

实例：n=4，m=3



解：X=<1,4,7,8>，最大连续区间为{1, …, 24}

回溯算法小结

(1) 适应于求解组合搜索问题（含组合优化问题）
(2) 求解条件：满足多米诺性质
(3) 解的表示：解向量，求解是不断扩充解向量的过程
(4) 回溯条件：
搜索问题-约束条件
优化问题-约束条件+代价函数
(5) 算法复杂性：最坏情况为指数，空间代价小
(6) 降低时间复杂性的主要途径：
利用对称性裁减子树
划分成子问题
(7) 分支策略（深度优先、宽度优先、宽深结合、优先函数）