



算法设计与分析—进阶篇

第五讲 再论搜索

哈尔滨工业大学
王宏志

wangzh@hit.edu.cn

<http://homepage.hit.edu.cn/pages/wang/>



本讲内容

5.1 剪枝方法论与人员安排问题

5.2 旅行商问题

5.3 A*算法

问题的定义

- 输入

- 人的集合 $P=\{P_1, P_2, \dots, P_n\}$,
 $P_1 < P_2 < \dots < P_n$, 工作的集合 $J=\{J_1, J_2, \dots, J_m\}$, J 是偏序集合

例. 给定 $P=\{P_1, P_2, P_3\}$, $J=\{J_1, J_2, J_3\}$, $J_1 \leq J_3$,
 $J_2 \leq J_3$.

$P_1 \rightarrow J_1$ 、 $P_2 \rightarrow J_2$ 、 $P_3 \rightarrow J_3$ 是可能的解.

$P_1 \rightarrow J_1$ 、 $P_2 \rightarrow J_3$ 、 $P_3 \rightarrow J_2$ 不可能是解.

工作

- 如果 $f(P_i) \leq f(P_j)$, 则 $P_i \leq P_j$

转换为树搜索问题

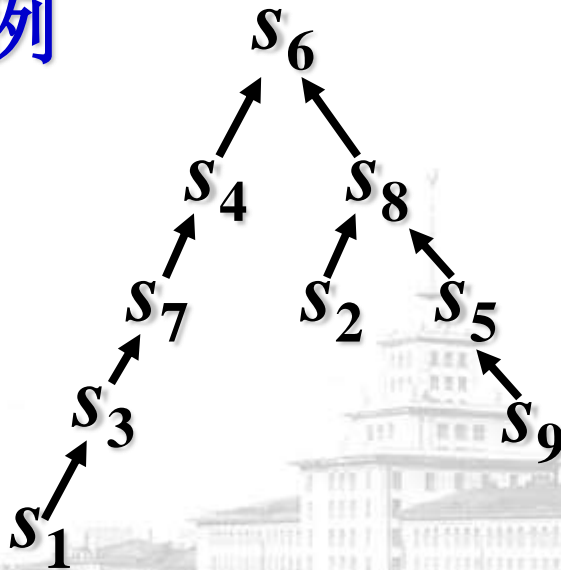
- 拓朴排序

- 输入: 偏序集合(S, \leq)

- 输出: S 的拓朴序列是 $\langle s_1, s_2, \dots, s_n \rangle$,
满足: 如果 $s_i \leq s_j$, 则 s_i 排在 s_j 的前面.

- 例

拓朴排序:



$s_1 s_3 s_7 s_4 s_9 s_5 s_2 s_8 s_6$

- 问题的解空间

命题1. $P_1 \rightarrow J_{k1}$ 、 $P_2 \rightarrow J_{k2}$ 、...、 $P_n \rightarrow J_{kn}$ 是一个可能解, 当且仅当 J_{k1} 、 J_{k2} 、...、 J_{kn} 必是一个拓朴排序的序列.

例 $P = (P_1, P_2, P_3, P_4)$ 、 $J = (J_1, J_2, J_3, J_4)$ 的值序如下

问题的解空间是所有拓朴排序的序列集合, 每个序列对应一个可能的解

例如 (J_1, J_2, J_3, J_4) 、 (J_1, J_2, J_4, J_3) 、 (J_1, J_3, J_2, J_4) 、 (J_2, J_1, J_3, J_4) 、 (J_2, J_1, J_4, J_3) 是拓朴排序序列

(J_1, J_2, J_4, J_3) 对应于 $P_1 \rightarrow J_1$ 、 $P_2 \rightarrow J_2$ 、 $P_3 \rightarrow J_4$ 、 $P_4 \rightarrow J_3$

-
- ```
graph TD; 0((0)) --> 1((1)); 0 --> 2((2)); 1 --> 2; 1 --> 3((3)); 2 --> 1; 2 --> 3; 2 --> 4((4)); 3 --> 2; 3 --> 4; 4 --> 3; 4 --> 4; 1 --> 4; 2 --> 3; 3 --> 1; 4 --> 2; 4 --> 1; 4 --> 0;
```

# ● 拓朴序列树的生成算法

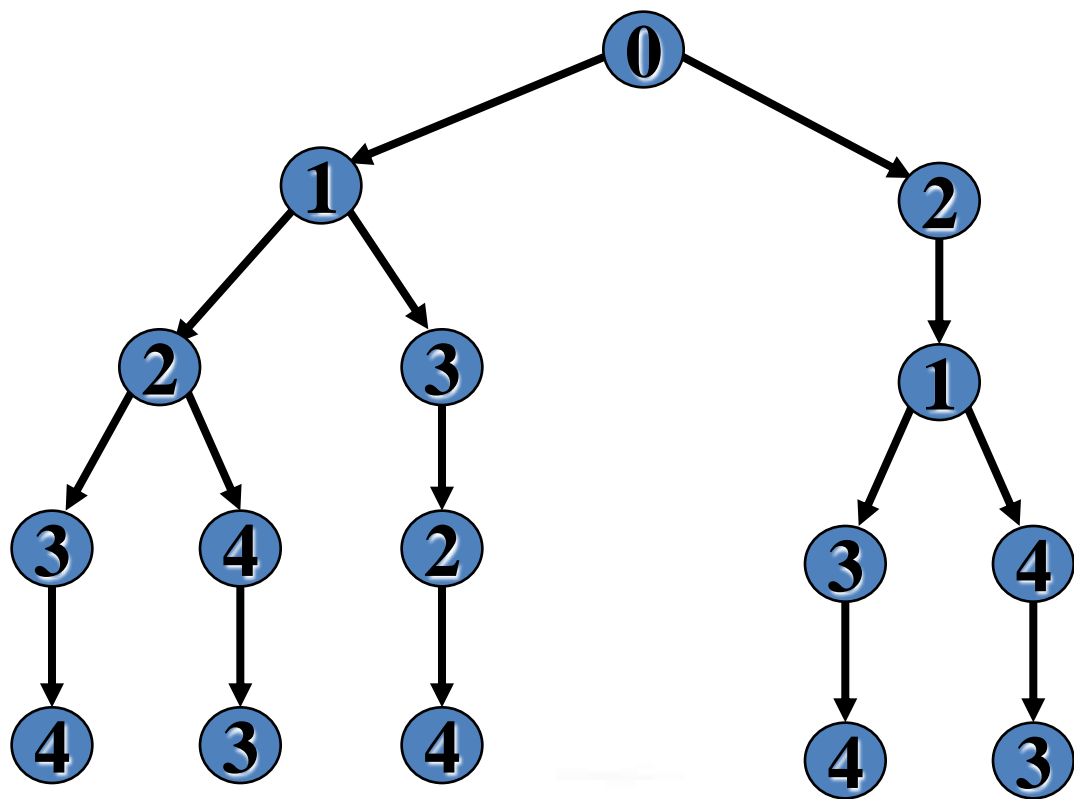
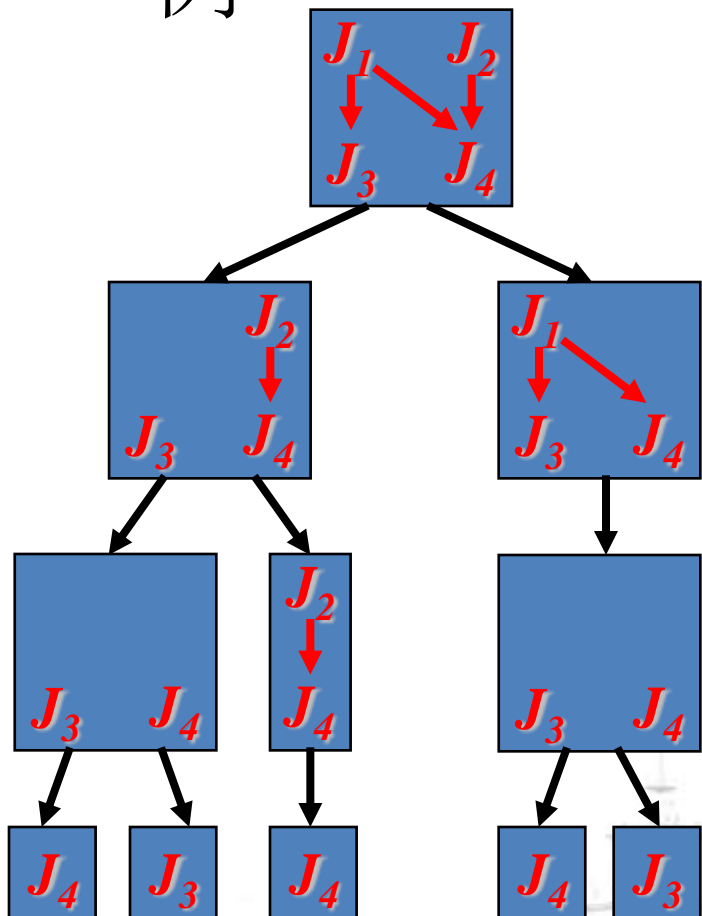
输入：偏序集合 $S$ , 树根 $root$ .

输出：由 $S$ 的所有拓朴排序序列构成的树.

1. 生成树根 $root$ ;
2. 选择偏序集中没有前序元素的所有元素, 作为 $root$ 的子节点;
3. **For**  $root$ 的每个子节点 $v$  **Do**
4.      $S = S - \{v\}$ ;
5.     把 $v$ 作为根, 递归地处理 $S$ .



• 例





# 求解问题的分支界限搜索 算法

- 计算解的代价的下界

**命题2.** 把代价矩阵某行(列)的各元素减去同一个数, 不影响优化解的求解.

- 代价矩阵的每行(列)减去同一个数(该行或列的最小数), 使得每行和每列至少有一个零, 其余各元素非负.
- 每行(列)减去的数的和即为解的下界.



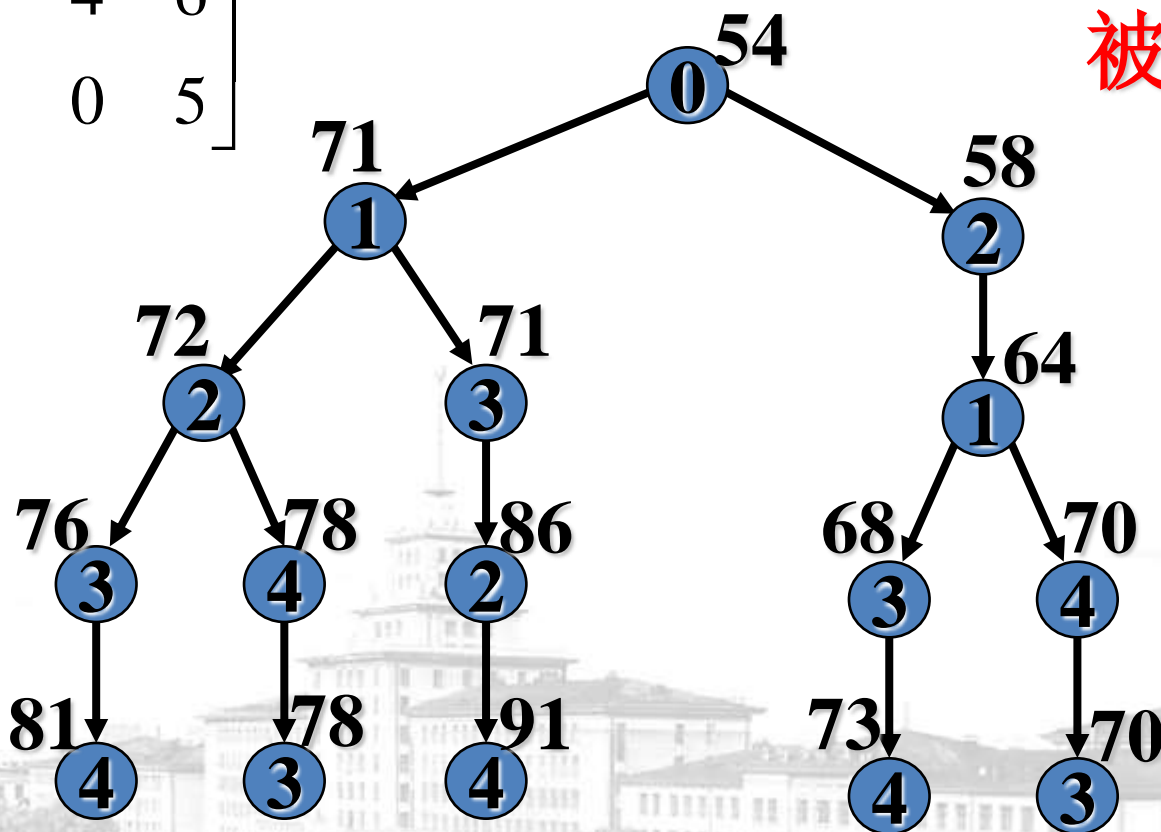
例.

$$\begin{array}{ccccc} & J_1 & J_2 & J_3 & J_4 \\ P_1 & \left[ \begin{array}{cccc} 29 & 19 & 17 & 12 \end{array} \right] & -12 \\ P_2 & \left[ \begin{array}{cccc} 32 & 30 & 26 & 28 \end{array} \right] & -26 \\ P_3 & \left[ \begin{array}{cccc} 3 & 21 & 7 & 9 \end{array} \right] & -3 \\ P_4 & \left[ \begin{array}{cccc} 18 & 13 & 10 & 15 \end{array} \right] & -10 \\ & & -3 & & \end{array} \quad \rightarrow \quad \begin{bmatrix} 17 & 4 & 5 & 0 \\ 6 & 1 & 0 & 2 \\ 0 & 15 & 4 & 6 \\ 8 & 0 & 0 & 5 \end{bmatrix}$$

解代价下界=12+26+3+10+3=54

# 解空间的加权树表示

|    |    |   |   |
|----|----|---|---|
| 17 | 4  | 5 | 0 |
| 6  | 1  | 0 | 2 |
| 0  | 15 | 4 | 6 |
| 8  | 0  | 0 | 5 |



被分配到的人

1

2

3

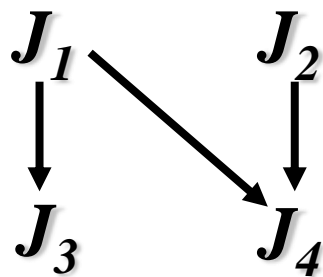
4

- 分支界限搜索(使用爬山法)算法
  1. 建立根节点,其权值为解代价下界;
  2. 使用爬山法,类似于拓朴排序序列树生成算法求解问题,每产生一个节点,其权值为加工后的代价矩阵对应元素加其父节点权值;
  3. 一旦发现一个可能解,将其代价作为界限,循环地进行分支界限搜索:剪掉不能导致优化解的解,使用爬山法继续扩展新增节点,直至发现优化解.



# • 例

$\{P_1, P_2, P_3, P_4\}$



|    |    |   |   |
|----|----|---|---|
| 17 | 4  | 5 | 0 |
| 6  | 1  | 0 | 2 |
| 0  | 15 | 4 | 6 |
| 8  | 0  | 0 | 5 |

