分枝-限界法

分枝(Branching)

- 分枝－一个节点成为E节点后,它要展开它的所有子节点;并将这些子节点放在一个称为活节点表的数据结构中;从活节点表中的节点可以展开所有状态空间树的节点,即广度优先遍历状态空间树.
- 按一定的规则从活节点表中取出一个节点作为E节点进行展开.
- 活节点表可以是FIFO, LIFO和优先级队列.
- 当使用优先级队列时必须对活节点表中的节点赋以一个权值.
- 本章结合求解优化问题介绍一种使用优先级队列的分枝限界法.

最小成本优化问题

- 设$x=(x_1, x_2, \ldots, x_n)$为可行解的元组;
- 对每个可行解有一个成本值，cost(x).
- 最小成本优化问题:

  求使cost(x)达到最小的可行解x.
- 使用搜索方法求解最小成本优化问题.

任意节点的成本函数c(x)

- 定义状态空间树上任一结点x的成本函数c(x)如下:
- 如x为可行叶结点则 c(x)=cost(x);
- 否则, 定义c(x)=从x 展开状态空间树能得到的最小成本值(状态空间树上以x为根的子树中可行解成本的最小值).
- 如其子树中无可行解则c(x)=∞

LC-检索

- 如果活节点表中每个节点以c(x)为权值,每次从活节点表中取出最小权值节点作为E-节点,则算法能很快找到优化解.
- 但在展开x 前不可能知道c(x)的值. 但是有可能从历史信息获得c(x)的某一下界ĉ(x).
- 以c(x)的下界估值ĉ(x)做为活节点表中节点的权值, 每次取出有最小ĉ(x)的节点进行展开;
- 要求设计的ĉ(x)满足:

  ĉ(x)=cost(x),当x为可行叶节点时;

基于优先级队列的分枝-限界算法

```
E=T;
置活节点表为空;
while(true)
{
  if E是可行叶节点 return;
  for E 的每个子节点x
       {Add(x); parent(x)=E}
       delete(E);
}
```

- delete(E)从活节点表中取出有最小$\hat{c}(x)$的活节点并从中删除之.
- 算法正确性证明：
  - 当算法结束在第4行时 $\hat{c}(E)=c(E)$;
  - 对活节点表中每个节点x, 都有:

    $$\hat{c}(x)\geq\hat{c}(E)=c(E)$$

  - 但$c(x)\geq\hat{c}(x)\geq c(E)$, 且$c(x)$为状态空间树上以x为根的子树的最小成本值.
  - 又因为活节点表覆盖了状态空间树所有叶节点, 所以$c(E)$是最小成本值, 而E是最优解.

```
E=T;
置活节点表为空;
  while(true)
{
  if E是可行节点且 ĉ(E)=cost(E) return;
  for E 的每个子节点x
        {Add(x); parent(x)=E}
        delete(E);
}
```

限界:

- 令U为当前获得的最优成本值;
- 设$x=(x_1,\ldots,x_k)$,如$\hat{c}(x)\geq U$,则停止展开子节点x, 即, 不将其放入活节点表.
- U初始为∞, 其后每次得到一个新的可行解, 用其成本值对U加以修改:

  U←min{U, cost(x)};

LC-分支-限界算法

E=T; U←∞;
置活节点表为空;
while(true)
{
for E 的每个子节点x
If x 是叶节点 then U←min{U,cost(x)};
if ĉ(x)<U then {Add(x), parent(x)=E;}
If 活节点表空 then 算法结束;
delete(E);
if ĉ(E)≥U 算法结束;
}

- 算法结束时, 如果$\hat{c}(E) \geq U$, 活节点表中其它节点x的下界也满足:

$$\hat{c}(x) \geq \hat{c}(E) \geq U,$$

  展开这些活节点不能产生更好的解.
- 算法结束时, U的值为优化值.
- 为加快搜索的速度, 在每个中间节点x处可进一步估计展开x能得到的优化成本值的某一上界u(x), 并使用

$$U \leftarrow \min\{u(x), U\}$$

  修改U的值.
- 从x展开得到的任何一个可行解都可作为u(x).

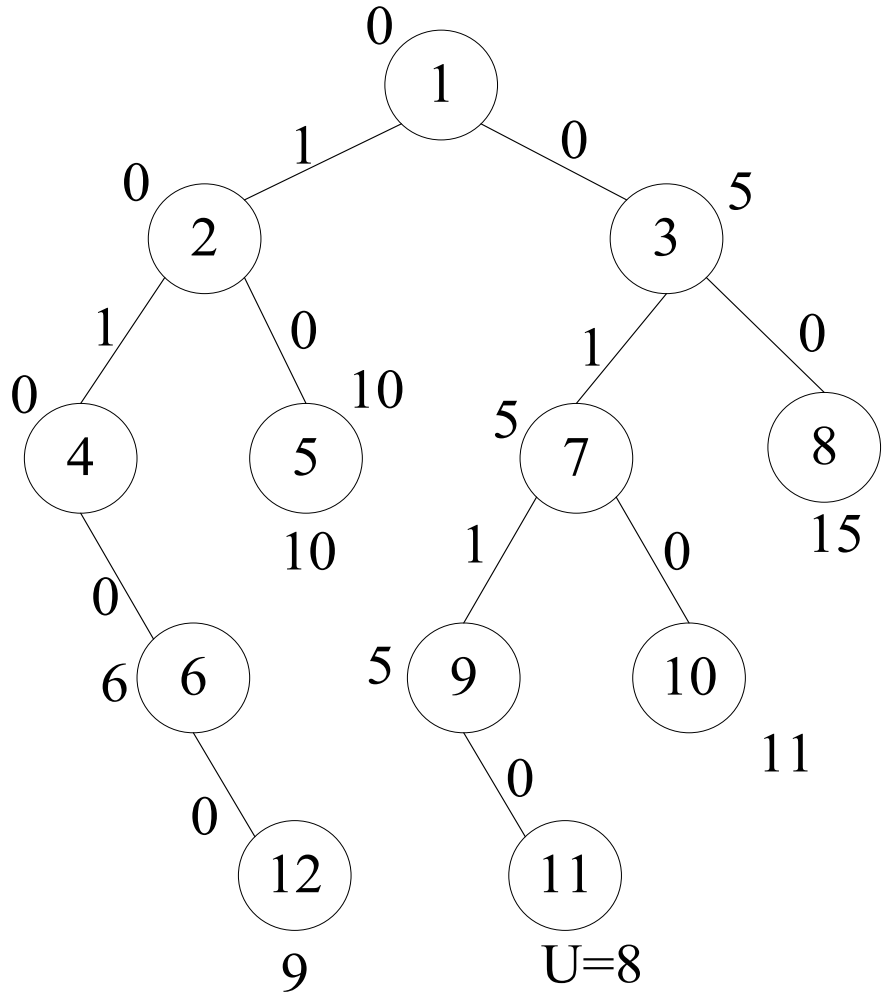
例1 带截止期的作业调度问题

- n个作业,1台处理机,每个作业i对应一个三元组(p_i,d_i,t_i)。
- p_i—罚款额;
- d_i—截止期;
- t_i —需要的处理机时间。
- 求可行的作业子集J,使得罚款额Σp_j最小,其中j为不在J中的作业.
- 定长元组表示可行作业子集:(x(1),---,x(n))
- 设X=(x(1),…x(k))为状态空间树的节点
- 下界$\hat{c}(x)$可估计为展开到x时已得到的罚款额:
  $\Sigma(1-x(j))p_j$,求和范围为$1\le j\le k$.

例1 带截止期的作业调度问题

- 已知4个作业的三元组(p_i,d_i,t_i)分别为(5,1,1), (10,3,2), (6,2,1), (3,1,1).
- 在每个结点x快速计算一个可行解, 并以该可行解的成本值,记为u(x), 修改U的值. 例如取x(k+1)=…x(n)=0,或用贪心法快速得到一个可行解.
- 算法每生成一个子节点时就用u(x)修改U

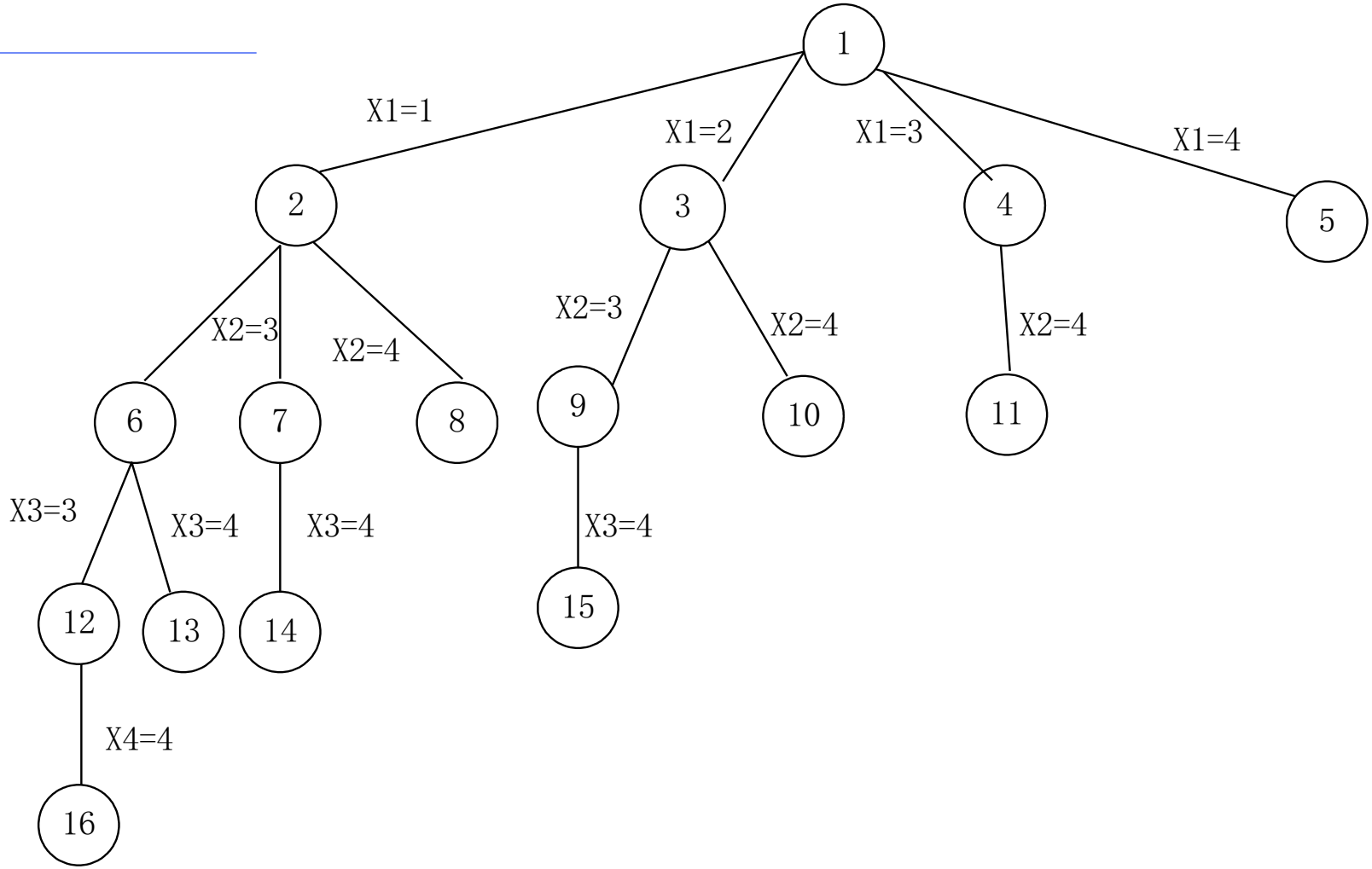
定长元组表示：LC-分枝—限界产生的部分状态空间树



子集问题的另一种状态空间树

- 设集合S=(x_1,---,x_n).S的一个子集A还可用A中元素在S中的下标作成的向量来表示.为使表示有唯一性, 限制按下标增序写这个向量.
- 例如A={x_1,x_3}可用向量(1,3)表示.向量(2)表示子集{x_2}.
- 一般情形我们用向量(j_1,---,j_k)表示子集:{x_{j1}, ---,x_{jk}}.要求j_1<---<j_k

调度问题的另一种状态空间树

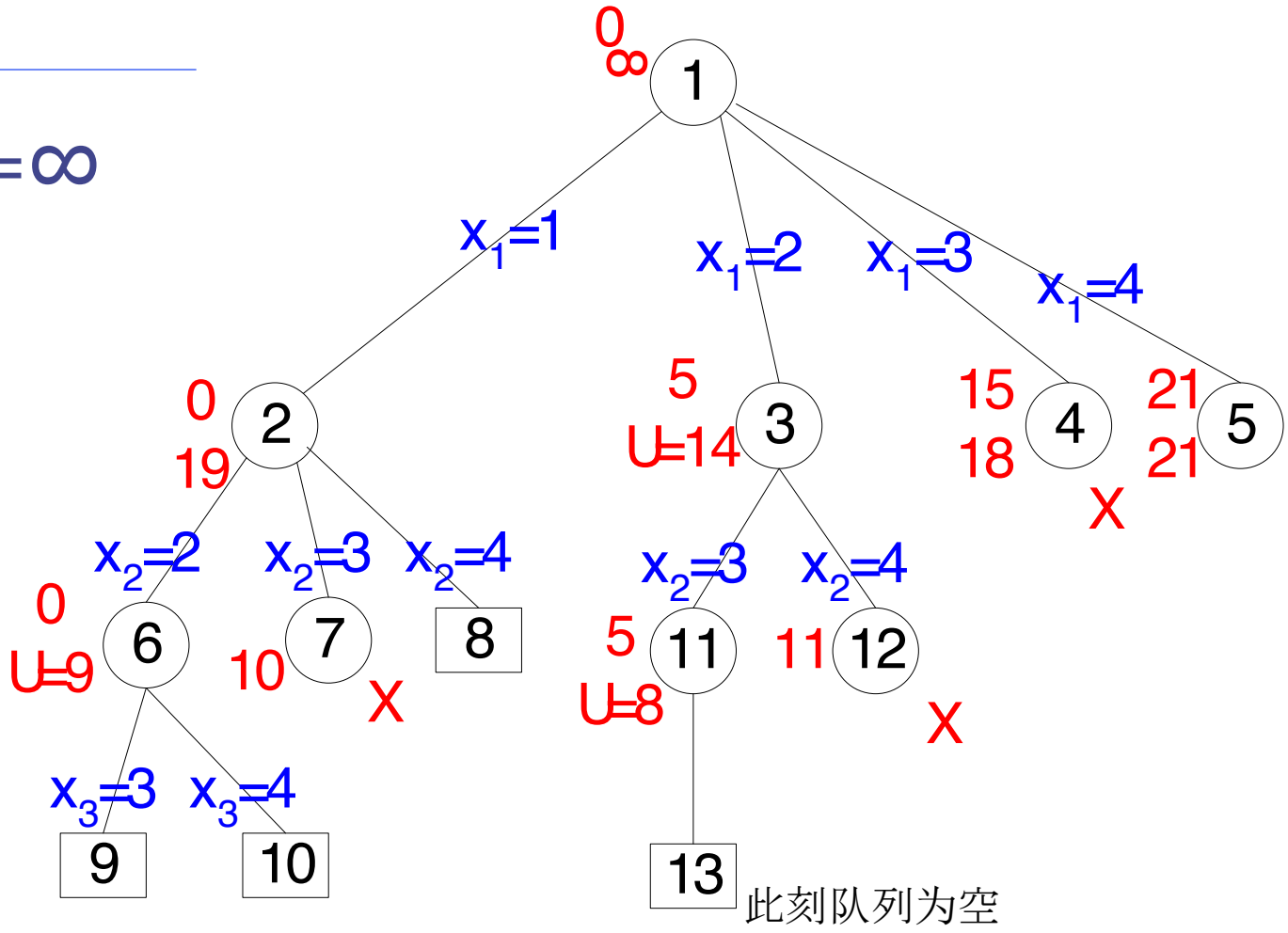


调度问题

- 可行条件(截至期)作为一种限界方法.
- ĉ(x)≥U为另一限界方法.
- 下页图中每个节点标注了2个数,上边的数为ĉ(x),下边的数为该节点对应的可行解的罚款额,作为u(x).
- 方框节点是非可行节点.
- 打叉的节点是被限界掉的节点.

LC-检索＋限界

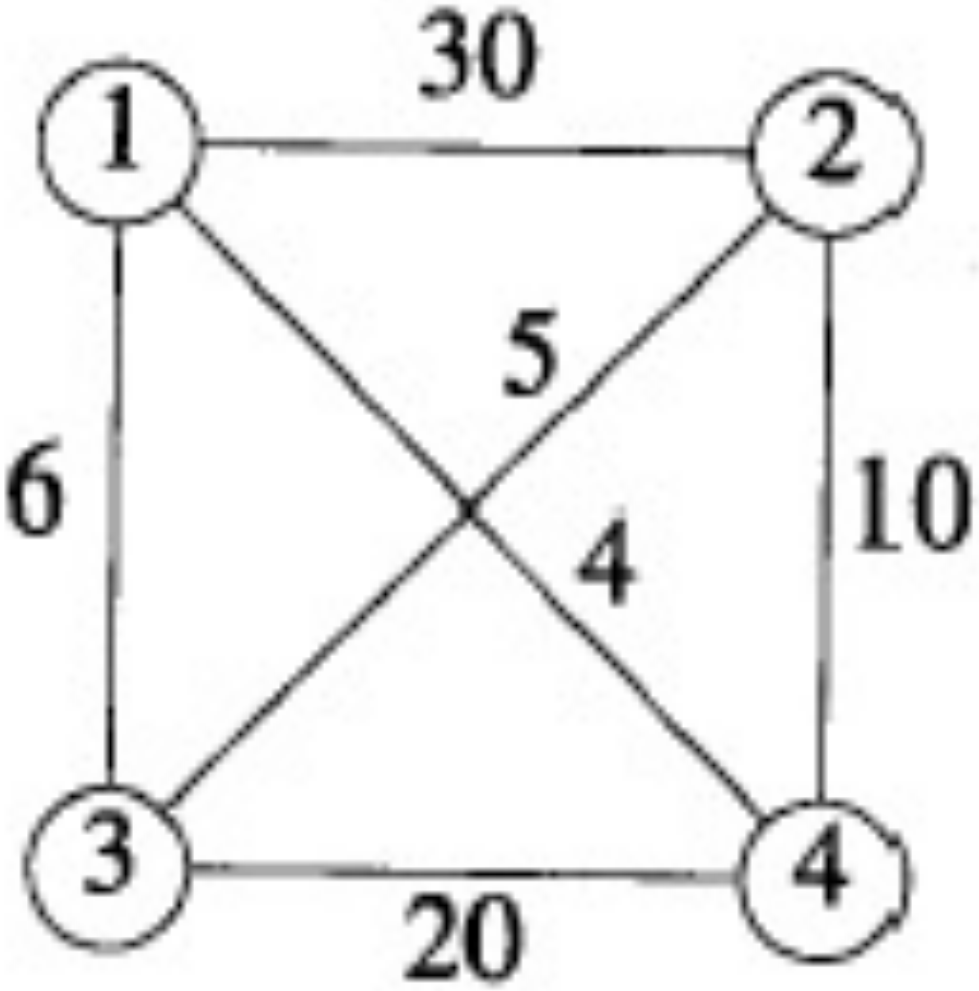
U=∞



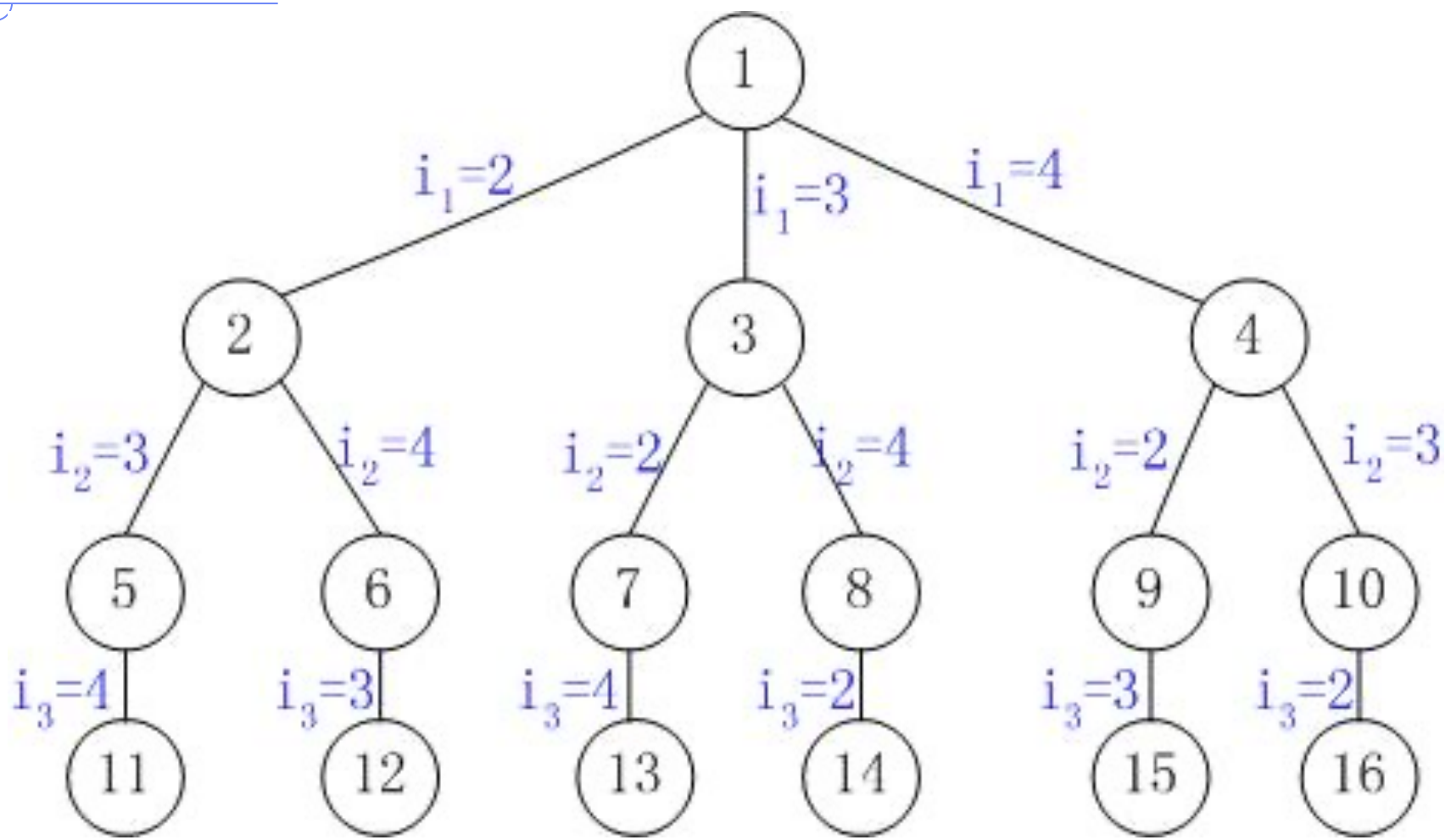
优化解为{2,3}，罚款额为8

货郎担问题

- 状态空间树
- 周游路线包括的边在邻接矩阵中不同行不同列
- 归约矩阵和归约数
- 归约数作为ĉ(X)


30
1
2
5
6
10
4
3
4
20

货郎担问题的状态空间树



一种计算ĉ(X)的方法

- 设f=(e_1,---, e_n)为一条周游路线:e_i来自邻接矩阵的第i行的边;所有e不同列.
- cost(e_i)=A(i,j_i), r_i=min{A(i,j)|1≤j≤n}
- Cost(f)≥$\Sigma_{1\le j\le n} r_j$; 称$\Sigma_{1\le j\le n} r_j$ 为行归约数.
- 设Ã(i,j)=A(i,j)-r_i ,则Ã为行归约后的矩阵
- Cost(f)=以Ã为成本矩阵时f的长度 $+\Sigma_{1\le j\le n} r_j$
- 对Ã做列归约得到矩阵R和归约数$\Sigma_{1\le j\le n} c_j$
- Cost(f)=f在R中的成本+ $\Sigma_{1\le j\le n} r_j$+ $\Sigma_{1\le j\le n} c_j$

归约矩阵和归约数

- 每行每列均含有0的矩阵称为归约阵
- 矩阵归约
- 假设第i行的约数为t_i,第j列的约数为r_j, $1 \le i, j \le n$,

  那么各行、列的约数之和 $L = \sum_{i=1}^{n} t_i + \sum_{j=1}^{n} r_j$ 称为矩阵约数

例

$$C=\begin{pmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{pmatrix}$$

归约矩阵和归约数

- 上述矩阵的行约数等于10＋2＋2＋3＋4=21
- 对右图矩阵做列归约得约数1＋0＋3＋0＋0=4
- 根节点的约数r=25
- ĉ(1)=25

$$C=\begin{pmatrix} \infty & 10 & 20 & 0 & 1 \\ 13 & \infty & 14 & 2 & 0 \\ 1 & 3 & \infty & 0 & 2 \\ 16 & 3 & 15 & \infty & 0 \\ 12 & 0 & 3 & 12 & \infty \end{pmatrix}$$

归约矩阵和归约数

- 对上一页矩阵归约得右图矩阵.
- 列归约数=4

$$C=\begin{pmatrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{pmatrix}$$

ĉ(X)计算

- 令S为R的子节点且S不是叶节点 ĉ(S)=ĉ(R)+A(i,j)+r,r为节点S处的归约数. A(i,j)为R的归约矩阵中边<i,j>的权值.
- 将R的归约矩阵中i行j列置为∞(禁止再选择节点i的出边和的j入边);将A(j,1)置为∞. 得到的S处的矩阵.对其归约得到r.
- 如S为叶节点,ĉ(S)=根到此叶节点的周游路线成本

ĉ(X)计算

- 节点2处的矩阵
- 已是归约矩阵
- ĉ(2)=25+10=35

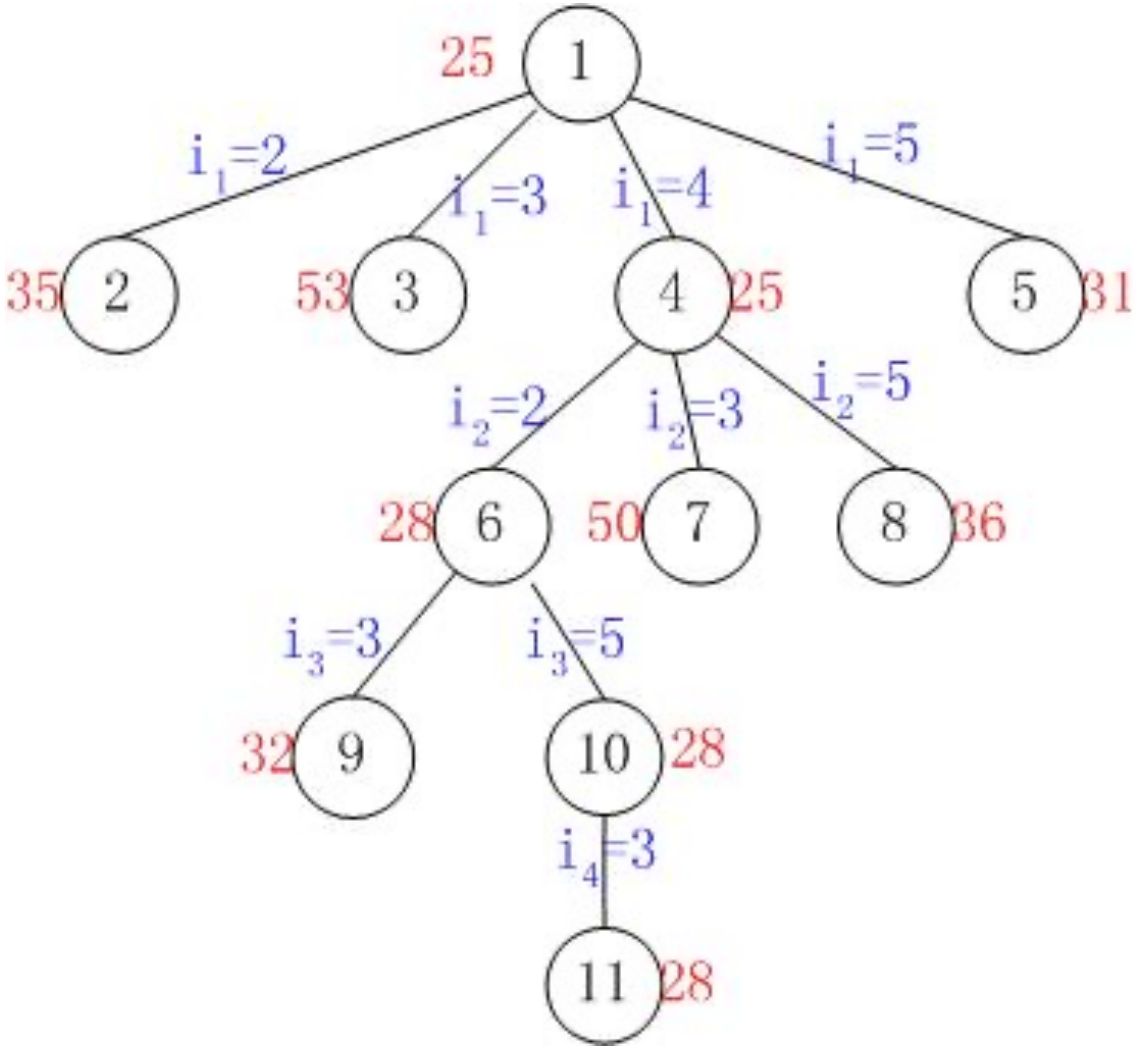
$$C=\begin{pmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 11 & 2 & 0 \\ 0 & \infty & \infty & 0 & 2 \\ 15 & \infty & 12 & \infty & 0 \\ 11 & \infty & 0 & 12 & \infty \end{pmatrix}$$

ĉ(X)计算

- 节点3处的矩阵
- 对其归约得归约数11
- ĉ(3)=25+17+11=53

$$C=\begin{pmatrix} \infty & \infty & \infty & \infty & \infty \\ 1 & \infty & \infty & 2 & 0 \\ \infty & 3 & \infty & 0 & 2 \\ 4 & 3 & \infty & \infty & 0 \\ 0 & 0 & \infty & 12 & \infty \end{pmatrix}$$

算法LCBB生成的状态空间树



17章习题

- 对以下最小罚款额调度问题的实例:
  (10,3,2),(3,4,2),(8,2,1),(6,3,1)
  分别用回溯法和基于LC-检索的分枝-限界法求解. 要求:写出限界条件;画出展开的部分状态空间树.
- 对以下0/1背包问题的实例:
  n=4,c=7,w=[3,5,2,1],p=[9,10,7,4]
  分别用回溯法和基于LC-检索的分枝-限界法求解. 要求:写出限界条件;画出展开的部分状态空间树.

- 考试时间:5月16日(12周周四)上午10点-12点
- 考试地点:等通知