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: 连通图$G=(V, E)$, 每个节点都没有到自身的边,每对节点之间都有一条非负加权边.

输出: 一条由任意一个节点开始
经过每个节点一次
最后返回开始节点的路径,
该路径的代价(即权值之和)最小.

转换为树搜索问题

- 所有解集合作为树根, 其权值由代价矩阵使用上节方法计算;
- 用爬山法**递归地**划分解空间, 得到二叉树
- 划分过程:
  - 选择图上边(i, j)使右子树代价下界增加最大
  - 所有包含(i, j)的解集合作为左子树
  - 所有不包含(i, j)的解集合作为右子树
  - 计算出左右子树的代价下界

分支界限搜索算法

- 在上述二叉树建立算法中增加如下策略:
  - 发现优化解的上界α;
  - 如果一个子节点的代价下界超过α, 则终止该节点的扩展.
- 下边我们用一个例子来说明算法

- 构造根节点, 设代价矩阵如下

$$
\begin{array}{ccc}
 & \begin{array}{ccccccc} j=1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} & \\
\begin{array}{c} i=1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} &
\left[\begin{array}{ccccccc}
\infty & 3 & 93 & 13 & 33 & 9 & 57 \\
4 & \infty & 77 & 42 & 21 & 16 & 34 \\
45 & 17 & \infty & 36 & 16 & 28 & 25 \\
39 & 90 & 80 & \infty & 56 & 7 & 91 \\
28 & 46 & 88 & 33 & \infty & 25 & 57 \\
3 & 88 & 18 & 46 & 92 & \infty & 7 \\
44 & 26 & 33 & 27 & 84 & 39 & \infty
\end{array}\right] &
\begin{array}{c} -3 \\ -4 \\ -16 \\ -7 \\ -25 \\ -3 \\ -26 \end{array} \\
 & \begin{array}{ccccccc} & & -7 & -1 & & & -4 \end{array} &
\end{array}
$$

- ➤ 根节点为所有解的集合
- ➤ 计算根节点的代价下界

- 得到如下根节点及其代价下界

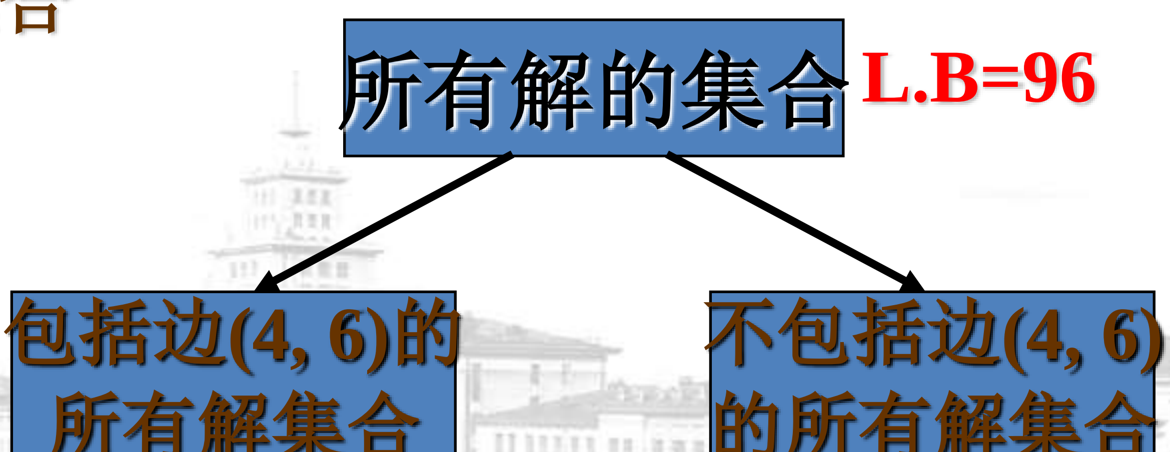
所有解的集合 **L.B=96**

- 变换后的代价矩阵为

$$
\begin{array}{c|ccccccc}
 & j=1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
i=1 & \infty & 0 & 83 & 9 & 30 & 6 & 50 \\
2 & 0 & \infty & 66 & 37 & 17 & 12 & 26 \\
3 & 29 & 1 & \infty & 19 & 0 & 12 & 5 \\
4 & 32 & 83 & 66 & \infty & 49 & 0 & 80 \\
5 & 3 & 21 & 56 & 7 & \infty & 0 & 28 \\
6 & 0 & 85 & 8 & 42 & 89 & \infty & 0 \\
7 & 18 & 0 & 0 & 0 & 58 & 13 & \infty
\end{array}
$$

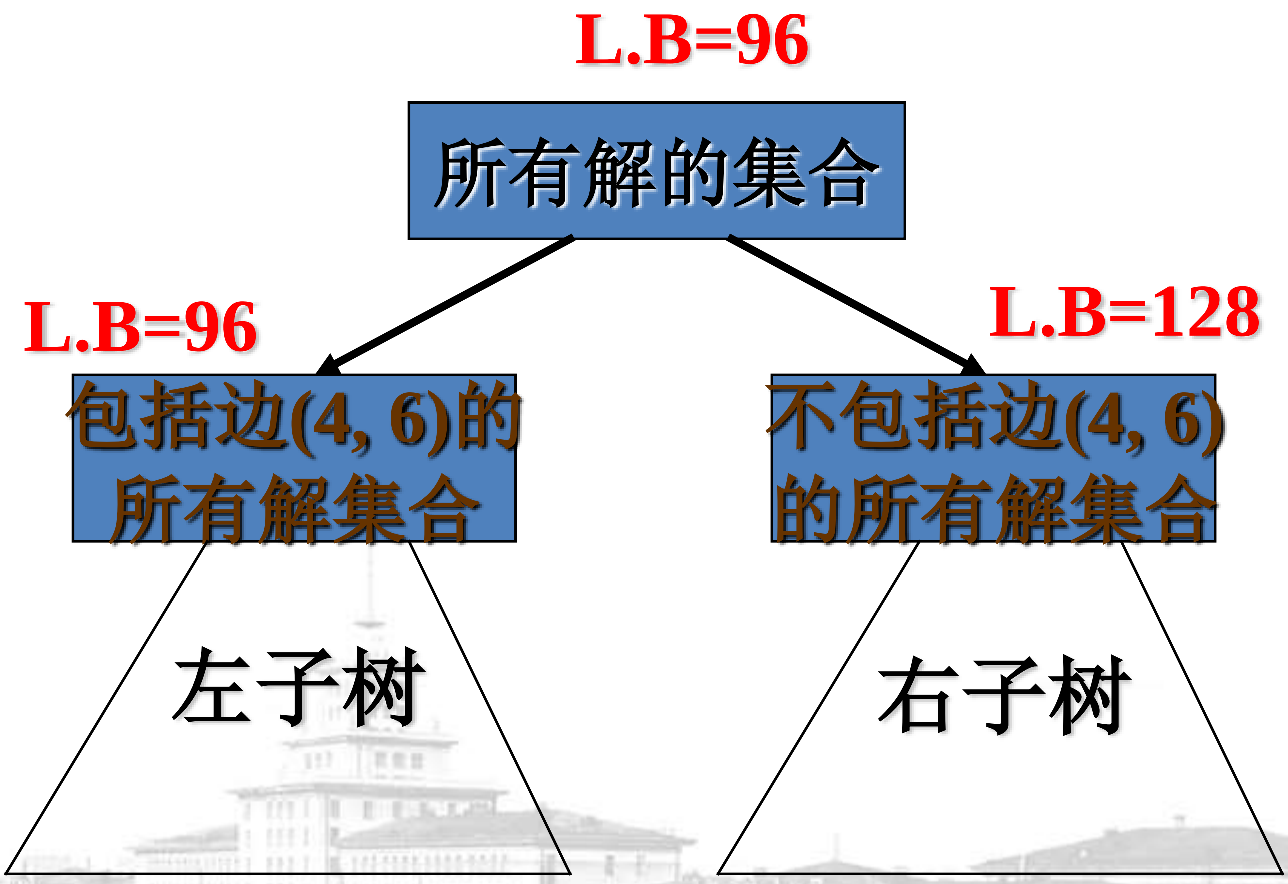
- 构造根节点的两个子节点
  - 选择使子节点代价下界增加最大的划分边(4, 6)
  - 建立根节点的子节点:
    - 左子节点为包括边(4, 6)的所有解集合
    - 左子节点为不包括边(4, 6)的所有解集合

$$
\begin{bmatrix}
\infty & 0 & 83 & 9 & 30 & 6 & 50 \\
0 & \infty & 66 & 37 & 17 & 12 & 26 \\
29 & 1 & \infty & 19 & 0 & 12 & 5 \\
32 & 83 & 66 & \infty & 49 & 0 & 80 \\
3 & 21 & 56 & 7 & \infty & 0 & 28 \\
0 & 85 & 8 & 42 & 89 & \infty & 0 \\
18 & 0 & 0 & 0 & 58 & 13 & \infty
\end{bmatrix}
$$



- 计算左右子节点的代价下界
  - (4, 6)的代价为0, 所以左节点代价下界仍为96.
  - 我们来计算右节点的代价下界:
    - 如果一个解不包含(4, 6), 它必包含一条从4出发的边和 进入节点6的边.
    - 由变换后的代价矩阵可知, 具有最小代价由4出发的边为(4, 1), 代价为32.
    - 由变换后的代价矩阵可知, 具有最小代价进入6的边为(5, 6)，代价为0.
    - 于是，右节点代价下界为: 96+32+0=128.

➢ 目前的树为



- 递归地构造左右子树
  - ➢ 构造左子树根对应的代价矩阵
    - ✓ 左子节点为包括边(4, 6)的所有解集合, 所以矩阵的第4行和第6列应该被删除
    - ✓ 由于边(4, 6)被使用, 边(6, 4)不能再使用, 所以代价矩阵的元素*C*[6, 4]应该设置为∞.
    - ✓ 结果矩阵如下

	j = *1*	*2*	*3*	*4*	*5*	*6*	*7*
i=	∞	0	83	9	30		50
1	0	∞	66	37	17		26
2	29	1	∞	19	0		5
4	3	21	56	7	∞		28
5	0	85	8	∞	89		0
6	18	0	0	0	58		∞
7							

➢ 计算左子树根的代价下界

- ✓ 矩阵的第5行不包含0
- ✓ 第5行元素减3，左子树根代价下界为:
  96+3=99
- ✓ 结果矩阵如下

$$
\begin{array}{c}
 \\ i=1 \\ 2 \\ 3 \\ 5 \\ 6 \\ 7
\end{array}
\begin{array}{c}
\begin{array}{cccccc} j=1 & 2 & 3 & 4 & 5 & 7 \end{array} \\
\begin{bmatrix}
\infty & 0 & 83 & 9 & 30 & 50 \\
0 & \infty & 66 & 37 & 17 & 26 \\
29 & 1 & \infty & 19 & 0 & 5 \\
0 & 18 & 53 & 4 & \infty & 25 \\
0 & 85 & 8 & \infty & 89 & 0 \\
18 & 0 & 0 & 0 & 58 & \infty
\end{bmatrix}
\end{array}
$$

➢ 构造右子树根对应的代价矩阵

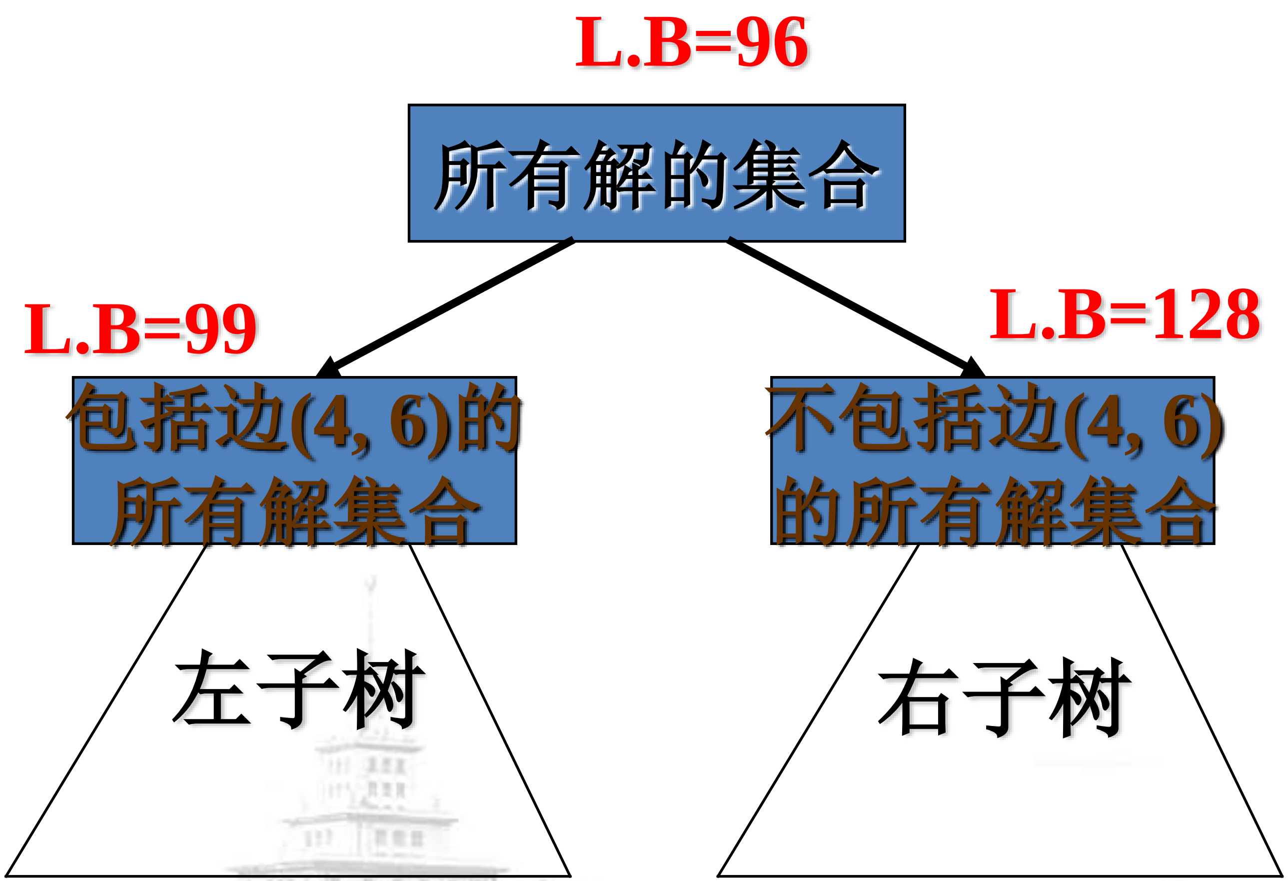
- ✓ 右子节点为不包括边(4, 6)的所有解集合, 只需要把 $C[4, 6]$设置为∞
- ✓ 结果矩阵如下

$$
\begin{array}{c c}
 & \begin{array}{ccccccc} j=1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\begin{array}{c} i=1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} &
\begin{bmatrix}
\infty & 0 & 83 & 9 & 30 & 6 & 50 \\
0 & \infty & 66 & 37 & 17 & 12 & 26 \\
29 & 1 & \infty & 19 & 0 & 12 & 5 \\
32 & 83 & 66 & \infty & 49 & \infty & 80 \\
3 & 21 & 56 & 7 & \infty & 0 & 28 \\
0 & 85 & 8 & 42 & 89 & \infty & 0 \\
18 & 0 & 0 & 0 & 58 & 13 & \infty
\end{bmatrix}
\end{array}
$$

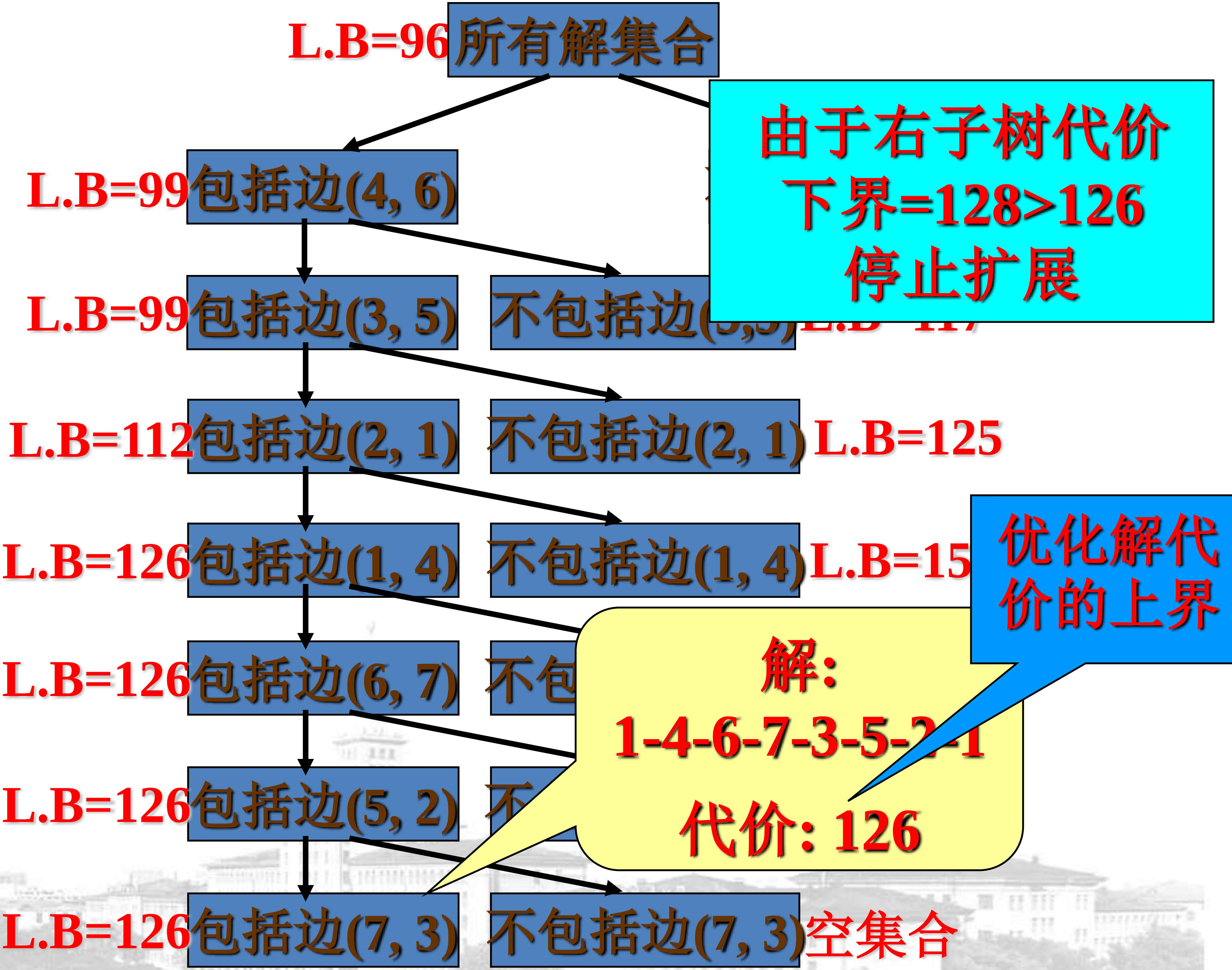
- 计算右子树根的代价下界
  - ✓ 矩阵的第4行不包含0
  - ✓ 第4行元素减32
  - ✓ 结果矩阵如下

$$
\begin{array}{c}
 \\
i=1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} j=1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\begin{bmatrix}
\infty & 0 & 83 & 9 & 30 & 6 & 50 \\
0 & \infty & 66 & 37 & 17 & 12 & 26 \\
29 & 1 & \infty & 19 & 0 & 12 & 5 \\
0 & 51 & 34 & \infty & 17 & \infty & 48 \\
3 & 21 & 56 & 7 & \infty & 0 & 28 \\
0 & 85 & 8 & 42 & 89 & \infty & 0 \\
18 & 0 & 0 & 0 & 58 & 13 & \infty
\end{bmatrix}
\end{array}
$$

➢ 目前的树为



- 使用爬山策略扩展左子树根
  - 选择边使子节点代价下界增加最大的划分边(3, 5)
  - 左子节点为包括边(3, 5)的所有解集合
  - 右子节点为不包括边(3, 5)的所有解集合
  - 计算左、右子节点的代价下界: 99和117
- 目前树扩展为:


L.B=96 所有解集合
L.B=99 包括边(4, 6)
L.B=99 包括边(3, 5)
不包括边(
L.B=112 包括边(2, 1)
不包括边(2, 1) L.B=125
L.B=126 包括边(1, 4)
不包括边(1, 4) L.B=15
L.B=126 包括边(6, 7)
L.B=126 包括边(5, 2)
L.B=126 包括边(7, 3)
不包括边(7, 3) 空集合
由于右子树代价
下界=128>126
停止扩展
优化解代价的上界
解:
1-4-6-7-3-5-2-1
代价: 126

注 意

如果i_1-i_2-…-i_m和j_1-j_2-…-j_m已被包含在一个正在构造的路径中, (i_m, j_1)被加入, 则必须避免j_n到 i_1的路径被加入. 否则出现环.