

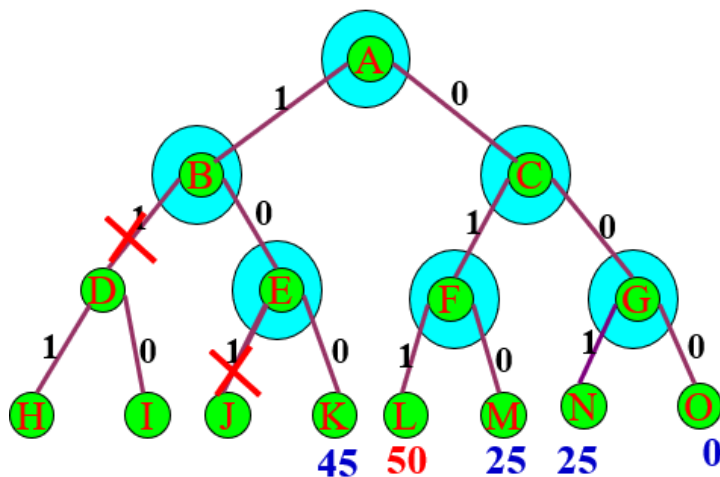
5.分支限界

广度优先搜索，最小耗费方法

- 队列式（FIFO）
- 优先队列式
- 搜索过程：活结点表

0-1背包

例 [0-1背包] $n=3$, $w=[20,15,15]$, $v=[45,25,25]$, $c=30$



活节点队列(FIFO策略)

[A]
[B C]
[C E]
[E F G]
[F G]
[G]
[]

A B C E F G

分支限界法-优先队列式

例1 $n=3$, $w=[20,15,15]$, $v=[45,25,25]$, $c=30$

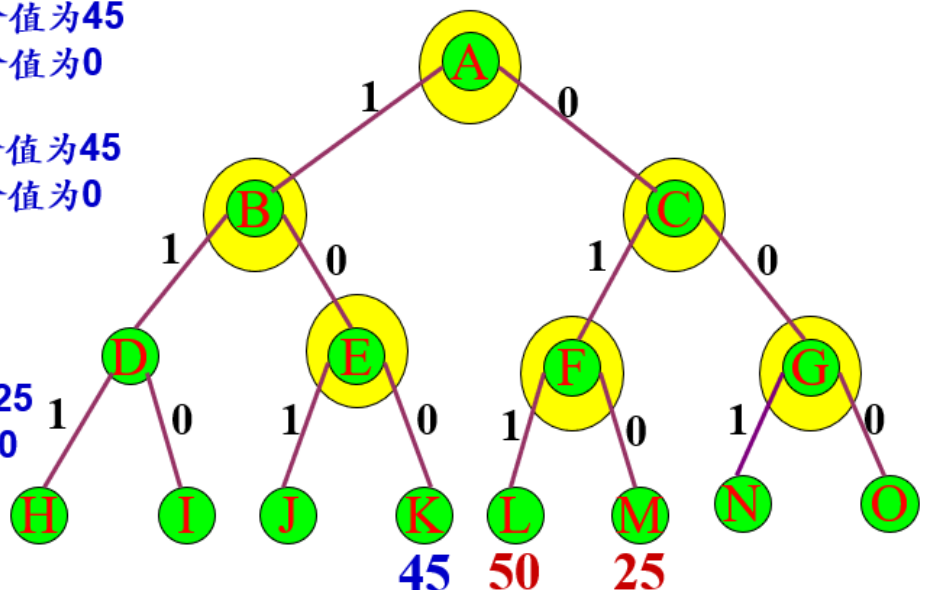
[B C] B-当前价值为45
C-当前价值为0

[E C] E-当前价值为45
C-当前价值为0

[C]

[F G] F-价值为25
G-价值为0

[G]



A B E C F G

不同点	回溯法	分支限界法
求解目标	找出树中满足约束条件的 所有解	找出满足约束条件的一个解或找出使目标函数达到 极大(小) 的最优解
搜索方式	深度 优先	广度 优先或 最小耗费 优先
扩展结点	多次 机会成为扩展结点: 扩展结点变为活结点后又可成为扩展结点	每个活结点只有 一次 机会成为扩展结点
树结点的生成顺序	生成 最近一个 有希望结点的单个子女	选择其中 最有希望 的结点,并生成它的所有子女
行进方向	随机性	方向性: 活结点表,搜索朝着解空间树上有最优解的分支推进

装载问题

• 可行性约束函数

$$\sum_{i=1}^n w_i x_i \leq c_1$$

ew: 子集树的第j+1层的节点Z处当前的装载重量, 即

$$ew = w_1 x_1 + w_2 x_2 + \cdots + w_j x_j$$

bestw: 当前最优载重量

r: 剩余集装箱的重量

• 上界函数

$$L = ew + r$$

已经装载的重量

剩余重量的上界

3、装载问题求解—队列式

$n = 4, c_1 = 12, c_2 = 8, W = \{8, 6, 3, 2\}$

活结点表

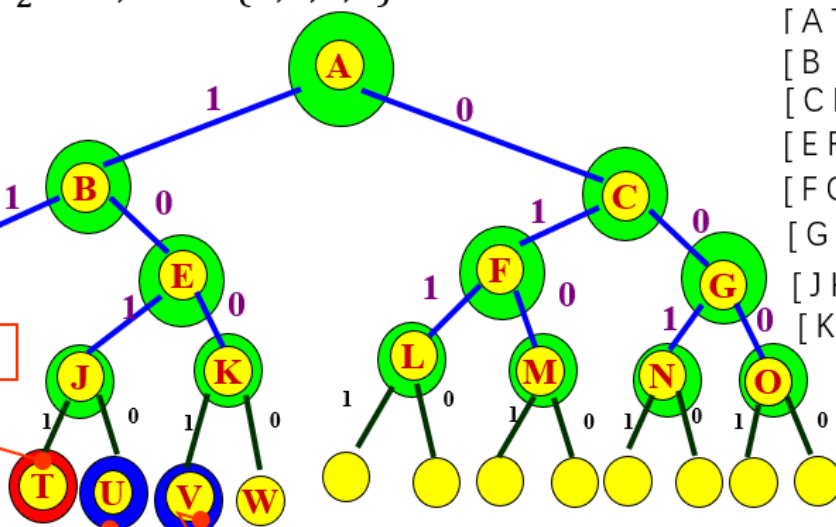
[A]
[B C]
[CE]
[EFG]
[FGJK]
[GJKLM]
[JKLMNO]
[KLMNO]
[LMNO]
[MNO]
[NO]
[O]
[]

ew=8
ew+w₂>12

不可行的叶结点

可行的叶结点
Bestw=11

可行的叶结点



3、装载问题求解—优先队列

- 用最大优先队列存储活结点表
- 优先级定义:

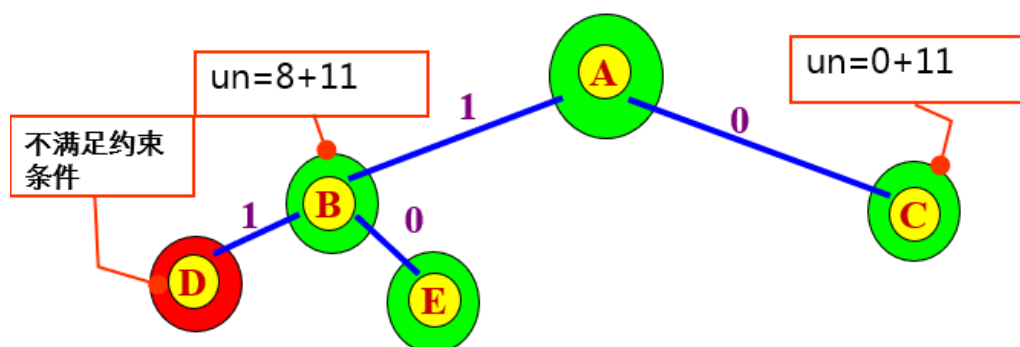
$$un = ew + r$$

- 优先队列中优先级最大的活结点成为下一个扩展结点

$$n = 4, c_1 = 12, W = \{8, 6, 3, 2\}$$

活结点表

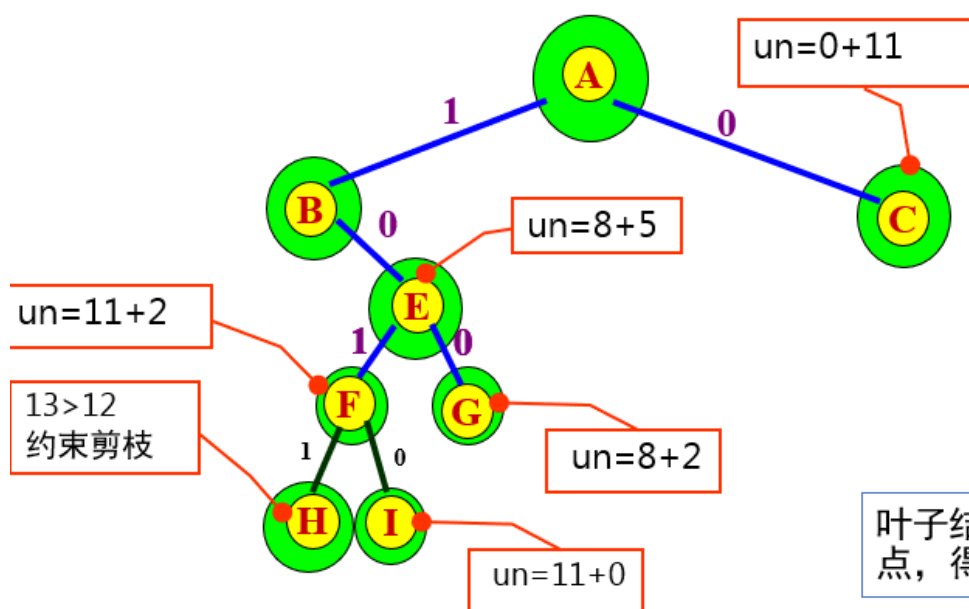
[A]
[B C]
[C E]



$$n = 4, c_1 = 12, W = \{8, 6, 3, 2\}$$

活结点表

[A]
[B C]
[C E]
[C F G]
[C G I]



叶子结点成为扩展结点，得到一个最优解