

用 MATLAB 编程构造最优二叉树^{*})

唐万梅

(重庆师范大学数计学院 重庆 400047)

摘 要 本文利用 MATLAB 编程实现了最优二叉树的构造,说明了基于 MATLAB 6.0 用动态规划法求解问题的方法。

关键词 二叉树,最优二叉树, MATLAB

1 引言

查找是数据结构中最基本的操作,其查找的方法很多。若查找表中元素按关键码有序且对任一元素的查找概率相等时,利用折半查找法进行查找,从而建立起相应的折半查找树(判定树)。按折半查找树描述的过程来进行折半查找,其性能(即平均查找长度)为最优,但如果有序表中各元素的查找概率不等时,又该如何进行查找,从而导致其最优性能呢?

例 假设有序表中含有四个记录,且已知各记录的查找概率不等,分别为 $p_1 = \frac{1}{4}$, $p_2 = \frac{1}{8}$, $p_3 = p_4 = \frac{1}{16}$ 的四个元素 $a_1 < a_2 < a_3 < a_4$, $q_0 = \frac{1}{8}$, $q_1 = \frac{3}{16}$, $q_2 = q_3 = q_4 = \frac{1}{16}$, 且 q_0 为查找那些 $\alpha < a_1$ 的 α 的概率, q_4 为查找 $\alpha > a_4$ 的概率,其余的 q_i 为查找 $a_i < \alpha < a_{i+1}$ 的概率,即 q_i 为查找失败时查找相应元素的概率。

为了定义二叉搜索树的耗费,一种可行的方法是给这棵二叉树附加 $n+1$ 片虚拟叶,用以表示 $n+1$ 种查找失败的情况,并把这些叶结点命名为 $0, 1, 2, \dots, n$ 。如图 1。

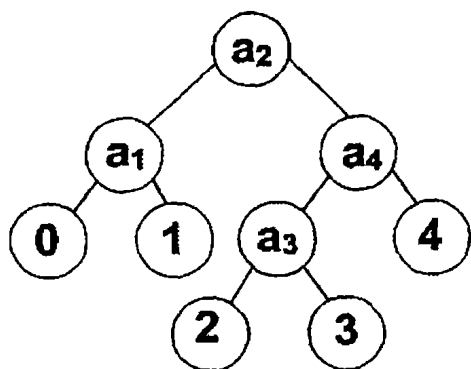


图 1

定义一棵二叉树的耗费为:

$$c_m = \sum_{i=1}^n p_i * \text{level}(a_i) + \sum_{i=0}^n nq_i * (\text{level}(i) - 1) \quad (1)$$

其中 $\text{level}(a_i)$ 为元素 a_i 所在层次; $\text{level}(i)$ 为编号为 i 的结点所在层次。

所谓最优二叉搜索树是一棵使(1)式取最小值的二叉搜索树。

如何求取最优二叉搜索树呢? 当然可用枚举方法进行: 对 $T_{02} = \{a_1, a_2\}$, 考察图 2。

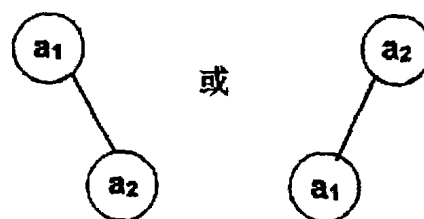


图 2

通过计算第一棵树的耗费为: $4 * 1 + 2 * 2 + 2 * 1 + 3 * 2 + 1 * 2 = 18$

第二棵树的耗费为: $2 * 1 + 4 * 2 + 2 * 2 + 3 * 2 + 1 * 1 = 21$, 即最好的为第一棵。以次类推。考察 $T_{03} = \{a_1, a_2, a_3\}, \dots, T_{04} = \{a_1, a_2, a_3, a_4\}$, 但显然这样的枚举法太浪费时间, 所以应改用其它方法。

2 用动态规划法求解

由于本问题满足最优化原理, 因此可用动态规划法来求解^[1]。

附最优性原理: 无论过程的初始状态和初始决策是什么, 其余的决策都必须相对于初始决策所产生的状态构成一个最优决策序列。

对 $S = \{a_1, a_2, \dots, a_n\}$, 假设 $a_1 < a_2 < \dots < a_n$, 概率 $q_0, q_i, p_i, 1 \leq i \leq n$ 的定义如前, 要求构造 S 的最小耗费二叉搜索树。

对 $0 < i < j < n$, 记 T_{ij} 是 S 的子集 $\{a_{i+1}, a_{i+2}, \dots, a_j\}$ 的最小耗费树, 而 C_{ij} 是 T_{ij} 的耗费值, r_{ij} 为 T_{ij} 的根, T_{ij} 的权 w_{ij} 定义为

$$\begin{aligned} W_{ij} &= q_i + (p_{i+1} + q_{i+1}) + \dots + (p_j + q_j) \\ &= W_{ij-1} + (p_j + q_j) \end{aligned}$$

^{*}) 重庆市运筹学与控制论重点实验室资助项目。

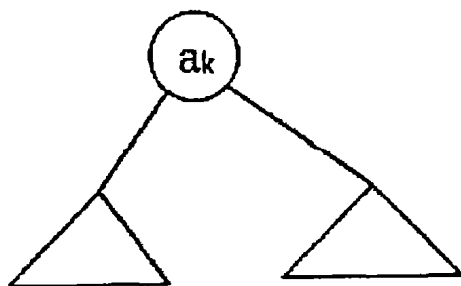


图 3

树 T_{ij} 由根 a_k (r_{ij} 的标记) 加上左子树 T_{ik-1} (它是 $\{a_{i+1}, a_{i+2}, \dots, a_{k-1}\}$ 的最小耗费树) 和右子树 T_{kj} (它是 $\{a_{k+1}, a_{k+2}, \dots, a_j\}$ 的最小耗费树) 所组成。如果 $i = k - 1$, 就没有左子树, 而如果 $k = j$, 就没有右子树, T_{ij} 看成空树, 它的权 W_{ii} 等于 q_i , 而它的耗费 $C_{ii} = 0$ 。

T 的耗费 C_{ij} 可表示为 $C_{ij} = W_{ik-1} + p_k + W_{kj} + C_{ik-1} + C_{kj} = W_{ij} + C_{ik-1} + C_{kj}$, 其中 k 是使 $C_{il-1} + C_{lj}$ 为最小的 $l(i < l \leq j)$ 。

用动态规划法, 对于 $0 \leq i < j \leq n$, 按 $j - i$ 的递增顺序计算出 W_{ij}, r_{ij} , 计算出各个 r_{ij} 后即可得出最终的最优二叉搜索树。

3 编程实现

用 MATLAB 6.0 编程实现 (说明: 由于 MATLAB 中数组、矩阵得下标均需从 1 开始, 所以, 在编程实现时将下标加 1, 计算出得最终结果要将下标减 1, 从而得出正确结果)。

附 MATLAB 程序如下;

```
w = [];
c = [];
r = [];
p = [0 4 2 1 1];
q = [2 3 1 1 1];
q(1);
for i = 1:4
    w(i,i) = q(i);
    r(i,i) = 0;
    c(i,i) = 0;
    w(i,i+1) = q(i) + q(i+1) + p(i+1);
    r(i,i+1) = i+1;
    c(i,i+1) = q(i) + q(i+1) + p(i+1);
end
```

```
w(5,5) = q(5);
r(5,5) = 0;
c(5,5) = 0;
for m = 2:5
    for i = 1:5-m
        j = i+m;
        w(i,j) = w(i,j-1) + p(j) + q(j);
        minn = c(i,i) + c(i+1,j);
        k = i+1;
        for l = i+2:j
            if minn >= (c(i,l-1) + c(l,j))
                minn = c(i,l-1) + c(l,j);
                k = l;
            end
        end
        c(i,j) = w(i,j) + c(i,k-1) + c(k,j);
        r(i,j) = k;
    end
end
disp(w)
disp(c)
disp(r)
```

根据程序计算出的矩阵 W, c, r , 构造出最后的最优二叉搜索树如图 4。

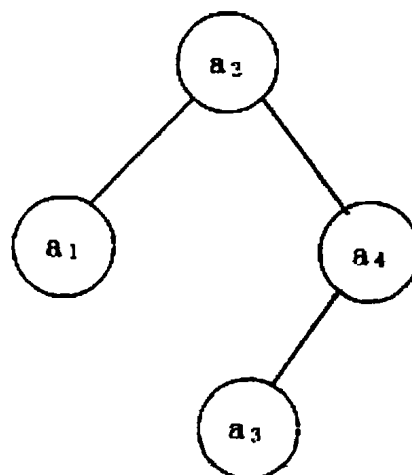


图 4

编程所得结果与实际结果相符。

结束语 通过本文说明了 MATLAB 6.0 在一类具体问题求解中的应用。

参考文献

- 1 余祥萱, 等. 计算机算法基础. 华中理工大学
- 2 刘宏友, 等. MATLAB 6.0 基础及应用. 重庆出版社