

浅析贪心算法

肖衡

(荆楚理工学院 计算机工程学院 荆门 448000)

摘要 本文讲述了贪心算法的含义,探讨并研究了贪心算法的基本思想及实现过程,通过实例分析了贪心算法的具体应用,指出了贪心算法的特点及存在问题。

关键词 贪心算法 贪心策略 贪心选择 最优子结构 最优解

中图分类号 TP3-05;TP301.6

文献标识码 A

文章编号 090905-5140

A Brief Analysis on Greedy Algorithm

Xiao Heng

(Department of Computer, Jingchu University of Technology Jingmen 448000)

Abstract This paper discusses on the meaning of greedy algorithm. Discussing its basic idea as well as studies its implementation process. This paper analyses ordinary applications of greedy algorithm through the actual cases, and points out its characteristics and existing issues.

Keywords Greedy algorithm Greedy policy Greedy choice Optimal sub-structure Optimal solution

一、贪心算法的含义

贪心算法可以简单描述为:对一组数据进行排序,找出最小值,进行处理,再找出最小值,再处理。也就是说贪心算法是一种在每一步选择中都采取在当前状态下最好或最优的选择,从而希望得到结果是最好或最优的算法。

贪心算法是通过一系列的选择来得到一个问题的解,而它所做的每一次选择都是当前状态下某种意义的最好选择,即贪心选择。即希望通过问题的局部最优解来求出整个问题的最优解。这种策略是一种很简洁的方法,对许多问题它能产生整体最优解,但不能保证总是有效,因为它不是对所有问题都能得到整体最优解,只能说其解必然是最优解的很好近似值。

二、贪心算法的基本要素

1、贪心选择

贪心选择是指所求问题的整体最优解可以通过一系列局部最优的选择,即贪心选择来达到。这是贪心算法可行的第一个基本要素,也是贪心算法与动态规划算法的主要区别。

贪心选择是采用从顶向下、以迭代的方法做出相继选择,每做一次贪心选择就将所求问题简化为一个规模更小的子问题。对于一个具体问题,要确定它是否具有贪心选择的性质,我们必须证明每一步所作的贪心选择最终能得到问题的最优解。

通常可以首先证明问题的一个整体最优解,是从贪心选择开始的,而且作了贪心选择后,原问题简化为一个规模更小的类似子问题。然后,用数学归纳法证明,通过每一步贪心选择,最终可得到问题的一个整体最优解。

2、最优子结构

当一个问题的最优解包含其子问题的最优解时,称此问题具有最优子结构性质。运用贪心策略在每一次转化时都取得了最优解。问题的最优子结构性质是该问题可用贪心算法或动态规划算法求解的关键特征。贪心算法的每一次操作都对结果产生直接影响,而动态规划则不是。贪心算法对每个子问题的解决方案都做出选择,不能回退,动态规划则会根据以前的选择结果对当前进行选择,有回退功能。动态规划主要运用于二维或三维问题,而贪心一般是一维问题。

三、贪心算法的思想及实现过程

1、贪心的基本思想

用局部解构造全局解,即从问题的某一个初始解逐步逼近给定的目标,以尽可能快地求得更好的解。当某个算法中的某一步不能再继续前进时,算法停止。贪心算法思想的本质就是分治,或者说:分治是贪心的基础。每次都形成局部最优解,换一种方法说,就是每次都处理出一个最好的方案。

2、贪心算法的实现过程

从问题的某一初始解出发；

while 能朝给定总目标前进一步 do

 求出可行解的一个解元素；

由所有解元素组合成问题的一个可行解

四、贪心算法的应用

贪心算法是计算机算法策略中常用的一个，往往在需要解决一些最优性问题时，都可以应用贪心算法。贪心法的常用范围有：一是明显的贪心，一般此类应用的题目本身就是贪心；二是贪心数据结构，如堆、胜者树；三是可证明贪心策略的贪心，这是我们最常见的；四是博弈、游戏策略，这些策略大多是贪心；五是求较优解或多次逼近最优解。

贪心算法的具体应用：以典型的背包问题为例。

给定 n 种物品和一个背包。物品 i 的重量是 W_i ，其价值为 V_i ，背包的容量为 C 。在选择物品 i 装入背包时，可以选择物品 i 的一部分，而不一定要全部装入背包， $1 \leq i \leq n$ 。应如何选择装入背包的物品，使得装入背包中物品的总价值最大？

根据题意，可以采用贪心算法来解决该问题，主要步骤为：首先计算每种物品单位重量的价值 V_i/W_i ，然后，依贪心选择策略，将尽可能多的单位重量价值最高的物品装入背包。若将这种物品全部装入背包后，背包内的物品总重量未超过 C ，则选择单位重量价值次高的物品并尽可能多地装入背包。依此策略一直地进行下去，直到背包装满为止。

使用贪心算法时，主要计算时间在于将各种物品依其单位重量的价值从大到小排序。因此，算法的计算时间上界为 $O(n \log n)$ 。同时为了证明算法的正确性，还必须证明背包问题具有贪心选择性质。

相关程序如下：

```
void Knapsack (int n,float M,float v[],float w[],float x[]) {
    Sort(n,v,w); // 各物品依单位重量的价值排序
    int i; float c=M;
    for (i=1;i<=n;i++) x[i]=0;
    for (i=1;i<=n;i++) {
        if (w[i]>c) break;
        x[i]=1;
        c -=w[i];
    }
    if (i<=n) x[i]=c / w[i];
}
```

五、贪心算法的特点及难点

贪心算法的最大特点就是快，通常是线性到二次式，不需要多少额外的内存。一般二次方级的存储要浪费额外的空间，

而且那些空间经常得不出正解。但是，使用贪心算法时，这些空间可以帮助算法更容易实现且更快执行。如果有正确贪心性质存在，那么一定要采用！因为它容易编写，容易调试，速度极快，并且节约空间。几乎可以说，此时它是所有算法中最好的。

但是应该注意，贪心算法有两大难点：

1、如何贪心

怎样用一个小规模的解构造更大规模的解呢？总体上，这与问题本身有关。但是大部分都是规律的。正因为贪心有如此性质，它才能比其他算法快。

2、贪心的正确性

要证明贪心性质的正确性，才是贪心算法的真正挑战，因为并不是每次局部最优解都会与整体最优解之间有联系，往往靠贪心算法生成的解不是最优解。这样，贪心性质的证明就成了贪心算法正确的关键。对某些问题贪心性质也许是错的，即使它在大部分数据中都是可行的，但还必须考虑到所有可能出现的特殊情况，并证明该贪心性质在这些特殊情况中仍然正确。而这样容易陷入证明不正确贪心性质的泥塘中无法自拔，因为贪心算法的适用范围并不大，而且有一部分极难证明，若是没有把握，最好不要冒险，还有其他算法会比它要保险。

六、结束语

贪心算法是很常见的算法，贪心策略是最接近人的日常思维的一种解题策略，虽然它不能保证求得的最优解一定是最佳的，但是它可以为某些问题确定一个可行性范围。贪心算法所作的选择依赖于以往所作过的选择，但决不依赖于将来的选择，这使得算法在编码和执行过程中都有一定的速度优势。对于一个问题的最优解只能用穷举法得到时，用贪心算法是寻找问题最优解的较好算法。对一个问题可以同时用几种方法解决，贪心算法并不是对所有的问题都能得到整体最优解或是最理想的近似解时，就需判断贪心性质的正确性了。与回溯法、动态规划法等比较，它的适用区域相对狭窄许多。

总之，如果一个贪心解决方案存在，就使用它。

参考文献

- [1] MH ALSUWAIYEL 算法设计技巧与分析[M]. 北京: 电子工业出版社(影印版), 51~52.
- [2] 朱洪. 算法设计和分析[M]. 上海: 科学技术文献出版社, 1989, 162~163.
- [3] 王晓东. 计算算法设计与分析(第二版)[M]. 北京: 电子工业出版社, 2004.
- [4] 王晓东. 算法设计与分析[M]. 北京: 清华大学出版社, 2004.
- [5] 苏德富, 钟诚. 计算机算法设计与分析[M]. 北京: 电子工业出版社, 2001, 60~62.

作者简介

肖衡(1979~)女，在读硕士，主要研究方向为计算网络。