

[文章编号] 1671-802X(2013)03-0078-03

# 贪心算法及其实例探究

杨书影

(安徽蚌埠第九中学, 安徽 蚌埠 233000)

**[摘 要]** 贪心算法就是在求最优解问题的过程中,依据某种贪心标准,从问题的初始状态出发,直接去求每一步的最优解,通过若干次的贪心选择,最终得出整个问题的最优解的方法。贪心算法所作的选择可以依赖于以往所作过的选择,但决不依赖于将来的选择,也不依赖于子问题的解,因此贪心算法与其它算法相比具有一定的速度优势。如果一个问题可以同时用几种方法解决,贪心算法应该是最好的选择之一。

**[关键词]** 贪心算法;贪心选择性质;时间复杂度

**[中图分类号]** TP301.6

**[文献标识码]** A

贪心算法模型是计算机算法学中解决各种现实问题的有效算法之一。当一个问题具有最优子结构性质和贪心选择性质时,贪心算法通常会给出一个更简单有效的解法。贪心算法并不从整体最优上加以考虑,它所作的选择只是在某种意义上的局部最优选择。

## 一、贪心算法概述

### (一)贪心算法的定义

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。

贪心算法是一种能够得到某种度量意义下的最优解的分级处理方法,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。

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。

利用贪心策略解题,需要解决两个问题:

- (1) 该题是否适合于用贪心策略求解;
- (2) 如何选择贪心标准,以得到问题的最优/较优解。

### (二)贪心算法的核心

贪心算法的核心问题是选择能产生问题最优解

\*[收稿日期] 2013-05-02

[作者简介] 杨书影(1978-),女,安徽蚌埠人,教师,研究方向:算法与程序设计。

的最优度量标准,即具体的贪心策略。

贪心策略是指从问题的初始状态出发,通过若干次的贪心选择而得出最优值(或较优解)的一种解题方法。其实,从“贪心策略”一词便可以看出,贪心策略总是做出在当前看来是最优的选择,也就是说贪心策略并不是从整体上加以考虑,它所做出的选择只是在某种意义上的局部最优解,而许多问题自身的特性决定了该题运用贪心策略可以得到最优解或较优解。

### (三)贪心算法的基本要素

#### 1.贪心选择性质

所谓贪心选择性质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。在动态规划算法中,每步所做的选择往往依赖于相关子问题的解。因而只有在解出相关子问题后,才能做出选择。而在贪心算法中,仅在当前状态下做出最好选择,即局部最优选择。然后再去解出这个选择后产生的相应的子问题。贪心算法则通常以自顶向下的方式进行,以迭代的方式做出相继的贪心选择,每做一次贪心选择就将所求问题简化为规模更小的子问题。

#### 2.最优子结构性性质

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。

#### (四)贪心算法的特点

贪心算法的最大特点就是快,通常是线性二次式,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。应该注意是,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。但是大部分都是有规律的。正因为贪心有如此性质,它才能比其他算法快。

具有应当采用贪心算法的问题,当“贪心序列”中的每项互异且当问题没有重叠性时,看起来总能通过贪心算法取得(近似)最优解的。或者,总有一种直觉在引导对一些问题采用贪心算法。经过实践发现,单纯的贪心算法是顺序处理问题的;而且每个结果是可以处理完一个数据后即时输出的。

#### (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。

#### (五)贪心算法存在的问题

1.不能保证求得的最优解是最佳的。由于贪心策略总是采用从局部看来是最优的选择,因此并不从整体上加以考虑。

2.贪心算法只能用来求某些最大或最小解的问题。

3.贪心算法只能确定某些问题的可行性范围。

### 二、贪心算法应用举例

#### (一)钓鱼问题

在一条水平路边,有  $n(2 \leq n \leq 25)$  个钓鱼湖,从左到右编号为  $1, 2, 3, \dots, n$ 。小明有  $H(1 \leq H \leq 16)$  个小时的空余时间,他希望用这些时间钓到尽量多的鱼。他从湖 1 出发,向右走,有选择的在一些湖边停留一定的时间钓鱼,最后在某一个湖边结束钓鱼。

小明测出从第  $i$  个湖到第  $i+1$  湖需要走  $5 \times T_i$  分钟的路,还测出在第  $i$  个湖边停留,第一个 5 分钟可以钓到鱼  $F_i$ ,以后再每钓 5 分钟鱼,鱼量减少  $D_i$ ,为了简化问题,小明假定没有其他人钓鱼,也不会有其他因素影响到他钓到期望数量的鱼。请求出能钓到最多鱼的方案。

【分析】每次选一个鱼最多的湖泊钓一次鱼。对于湖泊来说,由于在任何时候鱼的数目只和小明在该湖里钓鱼的次数有关,和钓鱼总次数无关,所以采用贪心策略比枚举法更合适。假设一共允许钓  $k$  次鱼,那么每次在  $N$  个湖泊中选择鱼最多的一个钓,选择每次钓鱼地点的时间复杂度为  $O(n)$ ,故总的时间复杂度为  $O(kn^2)$ 。

由此可以看到贪心法是每次选择一个局部最优策略进行实施,而不去考虑对今后的影响。一般来说,它的时间复杂度比较低,算法实现也比较容易。

## (二)汽车加油问题

一辆汽车加满油后,可行使  $n$  千米。旅途中有若干个加油站。若要使沿途加油次数最少,设计一个有效算法,对于给定的  $n$  和  $k$  个加油站位置,指出应在哪些加油站停靠加油才能使加油次数最少。

【分析】把两加油站的距离放在数组中, $a[1..k]$ 表示从起始位置开始跑,经过  $k$  个加油站, $a[i]$ 表示第  $i-1$  个加油站到第  $i$  个加油站的距离。汽车在运行的过程中如果能跑到下一个站则不加油,否则要加油。

对于这个问题有以下几种情况:设加油次数为  $k$ ,每个加油站间距离为  $a[i]; i=0,1,2,3 \dots n$

(1)始点到终点的距离小于  $N$ ,则加油次数  $k=0$ ;

(2)始点到终点的距离大于  $N$  时:

A.加油站间的距离相等,即  $a[i]=a[j]=L=N$ ,则加油次数最少  $k=n$ ;

B.加油站间的距离相等,即  $a[i]=a[j]=L>N$ ,则不可能到达终点;

C.加油站间的距离相等,即  $a[i]=a[j]=L<N$ ,则加油次数  $k=n/N(n\%N==0)$ 或  $k=[n/N]+1(n\%N \neq 0)$ ;

D.加油站间的距离不相等,即  $a[i] \neq a[j]$ ,则加油次数  $k$  通过贪心算法求解。

该题目求加油最少次数,即求最优解的问题,可分成几个步骤,不断地将问题实例归纳为更小的相似的子问题,并期望做出的局部最优的选择产生一个全局得最优解。

由于汽车是由始向终点方向开的,最大的麻烦就是不知道在哪个加油站加油可以使既可以到达终点又可以使加油次数最少。

提出问题是解决的开始。为了着手解决遇到的困难,取得最优方案。可以假设不到万不得已不加油,即除非油箱里的油不足以开到下一个加油站,才加一次油。在局部找到一个最优的解。每加一次油可以看作是一个新的起点,用相同的递归方法进行下去。最终将各个阶段的最优解合并为原问题的解得到原问题的求解。

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。

## [参考文献]

- [1] M.H.ALSUWAIYEL. 算法设计技巧与分析[M].北京:电子工业出版社,2004.
- [2] 王晓东.计算机算法设计与分析[M].北京:电子工业出版社,2001.
- [3] 常友渠.贪心算法的探讨与研究[J].重庆电力高等专科学校学报,2008,(9).
- [4] 刘汝佳,黄亮.算法艺术与信息学竞赛[M].北京:清华大学出版社,2006.

(责任编辑:卓如)