

早期邮票问题算法的优化 The Optimization of the Algorithm on Postage Stamps Problem

彭麟皓

PENG Lin-hao

(南宁市第二中学, 广西南宁 530012)
(No. 2 Middle School of Nanning, Nanning, Guangxi, 530012, China)

摘要:在分析早期邮票问题算法思路的基础上, 提出静态搜索限制规划、可变上界式动态搜索限制规划和可变上、下界式动态搜索限制规划对早期邮票问题算法进行优化. 优化后的算法在 $h=3, n=9$ 时计算邮票问题的大概时间分别为 13h, 6min, 11s. 动态搜索限制规划优化后的算法大大缩短了邮票问题的计算时间, 算法效率明显提升.

关键词: 动态规划 剪枝 可变下界 NP 问题

中图分类号: O112 **文献标识码:** A **文章编号:** 1002-7378(2008)02-0092-03

Abstract: Based on the analysis of early development problem in Stamps, this article proposed the optimizing algorithm by using static programming, dynamic programming with maximum value limitation only, and dynamic programming with minimum and maximum limitation. When $h = 3, n = 9$, the time for the calculation decreased to 11h 6min and 11s. The optimized dynamic programming significantly increased the efficiency of the algorithm.

Key words: dynamic programming, pruning, changeable limitation of minimum, no-polynomial problem.

邮票问题(或称为连续邮资问题)就是以有序集合 $A\{a_1, a_2, \dots, a_n\}$ (n 为自然数) 作为邮票的面值, 贴在一个最多能贴 h 张邮票的信封上, 确定一个最小的整数 k , 使得它不能被线性组合 $\sum_{i=1}^n X_i A_i, X_i > 0, \sum_{i=1}^n X_i < h$ 表示, 也就是说按顺序排列出第一个不能在这个信封上由 $A\{a_1, a_2, \dots, a_n\}$ 组合得出的邮票面值^[1].

对于邮票问题, 前人已经做了不少的研究, 例如: 当 $n=2$ 或 $n=3$ 时, 邮票最大面值可以被公式计算出来.

当 $n=2$ 时, $n_h(2) = (h+3-a_2)a_2 - 2$, 其中,

若 $h \geq a_2 - 2$, 那么 $n_h = [\frac{1}{4}(h^2 + 6h + 1)]$, $[x]$ 表示不超过 x 的最大整数; 当 $n=3$ 时, Hofmeister^[2] 找到适用于 $h \geq 20$ 时的公式 $n_h(3) = \frac{4}{3}(\frac{1}{3}h)^3 + 6(\frac{1}{3}h)^2 + Ah + B$, 其中 A, B 是函数 $f_x(h \bmod 9)$; 对于 $n=4$, Mossige^[3] 证明 $n_h(4) \geq 2.008(\frac{1}{4}h)^4 + O(h^3)$. 但是上述公式仅仅只能获得邮票面值的最大值, 而不能获得构成最大值的有序集合 $A\{a_1, a_2, \dots, a_n\}$. Shallit^[4] 证明了连续邮资问题是一个 NP 问题, 如果 k 给定, 则该问题时间复杂度可以被多项式表达. 本文在早期邮票问题算法的基础上, 对算法进行优化, 从而设计出更有效的算法, 尽可能减少时间复杂度, 为减少 NP 问题时间复杂度提供帮助.

1 早期邮票问题算法思路

在不考虑动态优化的情况下, 20 世纪 90 年代初期, 以大型计算机来运算邮票问题的思路一般都

收稿日期: 2008-02-25

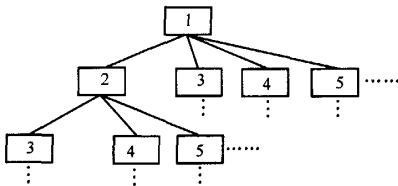
修回日期: 2008-04-05

作者简介: 彭麟皓(1991-), 男.

是确定一个明知道达不到的最大范围以确保搜索的完整性,并满足条件:对于每一次递归,栈中不出现重复元素.算法的结果输出必定是顺序的.

执行深度优先搜索,从深度为 1 开始递归执行到深度为 m .生成所有可能的结果,结果总个数大约为 \max^n ,因此其时间复杂度接近 $O(n^h)$.验证每个结果并计算每个结果得到的最大连续值,该原始算法复杂度为 $O(h^3)$,这种方法被称之为暴力搜索法.不使用 h 作为深度变量的原因在于, n 种邮票中到底有哪些“种”(邮票面值)是不确定的.如果采用 h 作为深度变量,每层深度中的 n 个种无法确定,而且对于加深的 h, n_{h-1}, n_h 间没有特别的联系.但是对于以 n 为深度变量的算法而言, h 总是恒定的.

暴力搜索法的生成树为



用伪代码模拟这个过程(以后的伪代码使用的字母和变量都是通用的).

读入数据 h 以及 n ;

设置最大搜索限制 $L_i = n^h$;

设置最大栈容量为 h ,初始化底层限制 $dlow = 0$;

{递归结构}

对 i 执行循环, i 从 $dlow + 1$ 到 L_i 中取值;

当前 i 进栈,如果栈指针指向栈顶,则计算连续邮资问题以栈中元素取值时为最大值;

如果这个最大值比 $kmax$ 大,将此值赋予 $kmax$,并保存当前栈状态退出本层递归;

否则,设置下层底层限制 $dlow$ 为 i ,递归进入下一层,栈指针上移一位;

循环结构完毕;

递归结构完毕;

输出 $kmax$,以及 $kmax$ 所对应的栈状态.

显然,这个算法的效率是很低的,对 $h = 3, n = 8$ 的计算时间就长达 6 个小时.

2 算法优化

2.1 静态搜索限制规划

在未使用动态思想的情况下,以 1 个对每次计算都需要调整的静态模型起步,确认每层递归搜索

限制的数学模型为

$$L_i = Ki + C(L_i \text{ 为第 } i \text{ 层时的最大搜索限制}).$$

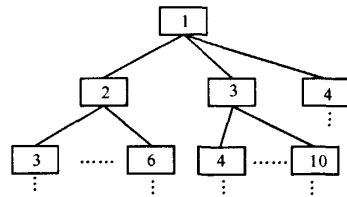
2.2 可变上界式动态搜索限制规划

使用动态规划思想,设计一个对应每个深度的最大限制,以避免无意义的搜索.

动态规划用于确认每层递归搜索限制的数学模型为 $L_i = f_{x,y}(i, h)$,其中 $f_{x,y} = \max$,对 \max 的定义如下:

定义一个多维向量 RX ,且 $RX_1 = 1, W_{x,y}(\max + 1, h) = \min(W_{x,y}(\max + 1 - K_a, h) + 1), RX_{\max+1} = 1, \max = \max + 1 \text{ s.t. } RX_{\max+1-k_2} = 1, W_{x,y}(\max + 1 - k_a, h) < h$,其中 K_a 为当前组合中第 a 个元素.

若 L_i 的定义同理 2.1 项,函数实际上以递归方式定义, $f_{x,y}$ 为以当前栈中数据所能得到的连续最大值, K_i 为目前的栈中元素, $W_{f_{x,y}}$ 记录此时已经占用了信封上几个格子,显然 $W_a \leq h$ 恒成立.算法的生成树为



该算法的伪代码描述为

读入数据 h 以及 n ;

定义栈指针变量为 p ,赋值为 1;

设置最大栈容量为 h ,初始化底层限制 $dlow_p = 1$;

将 1 压入栈, p 赋值为 2;

{递归结构}

对当前栈中元素进行运算,获得一个最大值,记录进 $dmax$;

对 i 执行循环, i 从 $dlow_p + 1$ 循环到 $dmax_p$;

当前 i 进栈;

如果栈指针指向栈顶,则计算连续邮资问题以栈中元素取值时的最大值;

如果这个最大值比 $kmax$ 大,将此值赋予 $kmax$,并保存当前栈状态,退出本层递归;

否则;

设置下层底层限制 $dlow_{p+1} = i$;

递归进入下一层, p 自加 1;

循环结构完毕;

递归结构完毕;

输出 $kmax$, 以及 $kmax$ 所对应的栈状态.

2.3 可变上、下界式动态搜索限制规划

对于确定了 1, 2, 3 的一个栈而言, 如果接下来 4 进栈, 显然也是没有意义的, 因为 1, 2 能够到达的最大值是 6, 如果需要让 4 进栈, 它完全可以取代现在 3 的位置以节省占用一个栈空间 ($n = 9$ 实际上起到限定栈最大空间为 9 的作用).

该动态规划用于确认每层递归搜索下限 —— 可变下界的数学模型为

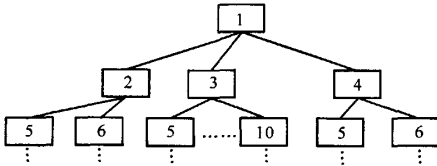
$$L_{\min,i} = L_{\max,i-1} + 1;$$

用于确认递归搜索上限数学的模型为

$$L_{\max,i} = f_{x,y}(i - 1, h).$$

函数 $f_{x,y}$ 定义同理 2.2 项.

限制下界以后, 该算法生成树为



限制下界以后的算法伪代码描述与 2.2 中算法相比较, 只需在“否则”后, 设置 $dlow_{p+1} = dmax_p$; 递归进入下一层, 栈指针上移一位即可.

3 结束语

静态搜索限制规划中的静态模型有致命的弱点, 即对限制的调节事实上是人为完成. 因此, 一方面有可能因为范围不够而没有得到正解, 而另一方

面又有可能因为范围过大而浪费时间. 当 $h = 3, n = 9$ 时, 按静态搜索限制规划改进算法后, 邮票问题的计算时间缩短到大约 13h. 按可变上界式动态搜索限制规划改进算法后, 邮票问题的计算时间缩短到大约 6min. 按可变上、下界式动态搜索限制规划改进算法后, 邮票问题计算时间大约 11s. 相对于早期算法思路, 本文主要的改进在于验证过程. 该过程用于计算每一组可能解的连续最大值, 将其变为动态过程, 数组中每个变量的计算都与其之前的量相关, 有效地降低重复运算所浪费的时间. 但是, 可变上、下界式动态搜索限制规划部分中的想法对于最后一次递归并不适用. 这有待于进一步完善.

参考文献:

[1] Richard A Brualdi. 组合数学[M]. 冯舜玺, 罗平, 裴伟东, 译. 北京: 机械工业出版社, 2004.

[2] Hofmeister G. Asymptotische abschätzungen für dreielementige extremalbasen in natürlichen Zahlen[J]. J reine angew Math, 1968, 232: 77-101.

[3] Svein Mossige. The postage stamp problem: an algorithm to determine the h-range on the h-range formula on the extremal basis problem for $k = 4$ [J]. Mathematics of Computation, 2000, 69(229): 325-337.

[4] Shallit J. The computational complexity of the local postage stamp problem[J]. Acm Sigact, 2002, 33: 90-94.

(责任编辑: 尹 闯 邓大玉)

世界最小晶体管问世

英国曼彻斯特大学研究人员诺沃舍罗夫带领的研究小组采用石墨烯材料制成目前世界上最小的电子晶体管, 其厚度为 1 个原子, 直径 10 个原子. 这种最新研制的新型电子晶体管比 32 纳米硅材料电子晶体管小 3 倍. 这种电子晶体管可用于任何半导体制造.

研究人员使用标准半导体制作工艺研制石墨烯晶体管, 他们使用一小片石墨烯, 并采用电子束平版印刷术雕刻出电路, 在石墨烯片中心保留着一个带有微小圆笼(circular cage)的量子点, 以电压改变量子点的传导率, 使它们像标准场效应晶体管的存储逻辑状态. 这种由石墨烯制成的晶体管由错综复杂的石墨烯碳原子构成, 具有很好的传导性能. 但是以当前的技术只能生产 100 微米直径的石墨烯晶体, 这一规格远小于 Intel 公司的工业生产标准. 但是科学家认为批量生产石墨烯晶体技术在未来是可以实现的.

(据科学网)