第19卷第4期 数学研究与评论 Vol. 19 No. 4
1999年11月 JOURNAL OF MATHEMATICAL RESEARCH AND EXPOSITION Nov. 1999

关于矩阵乘法的一个改进算法的时间复杂度*

张 振 祥[1,2]

(1. 安徽师范大学数学系，芜湖 241000；
2. 中国科技大学研究生院信息安全国家重点实验室，北京 100039)

摘 要：两个 n 阶非负整数方阵相乘，常规算法的时间复杂度为 $O(n^3)$，文献[1]提出一个“运算次数”为 $O(n^2)$ 的“最佳”算法，文献[2]对此算法做了进一步研究，提出三种改进策略. 本文根据算法分析理论，得出改进后的算法的时间复杂度仍不低于 $O(n^3\log n)$，因而其阶仍高于常规算法的运算量的阶.

关键词：矩阵乘法；算法分析；比特运算次数；时间复杂度；计算数论.
分类号：AMS(1991) 65F, 11Y16/CLC O151.21
文献标识码：A **文章编号**：1000-341X(1999)04-0716-03

1 引 言

两个 n 阶非负整数方阵(矩阵元素的最大值都不超过与 n 无关的某个常数) 相乘，常规算法的时间复杂度为 $O(n^3)$([3] 例 1 的推论)，Strassen 算法(1969) 的时间复杂度为 $O(n^{\log_2 7})$. 文献[1] 提出一个“运算次数” 为 $O(n^2)$ 的算法，作者称之为“达到理论下界”，是“最佳” 算法. 先[4] 后[5] 根据算法分析理论，用比特运算次数的概念，得出此算法的时间复杂度不低于 $O(n^3\log n)$ 和 $O(n^4\log n)$. [3,4,5,6] 多次指出，文献[1] 出错的原因在于忽略了他们算法中的每次运算(乘法，加法和除法) 耗时是问题规模(即矩阵的阶 n) 的非常值函数这个事实.

文献[2]显然没有注意到此项工作，并坚持认为文[1]的算法运算量为 $O(n^2)$，并提出三种改进策略，称“按这种思想设计的矩阵乘法算法不只是理论上的意义，同时向实际应用迈出了可喜的一步，”文[2]还给出了阶数≤100 的算例结果分析图，让人相信他们算法的有效性. 然而从此分析图看到 Strassen 算法比常规算法更耗时，但 Strassen 算法的时间复杂度低于常规算法的时间复杂度是已得到包括文[1,2]自己在内的普遍承认的，文[2]对此自相矛盾的现象没有作任何解释.

本文第二节得出文[2]的改进算法的时间复杂度仍不低于 $O(n^3\log n)$，第三节判析文[2] 的算例结果分析图，对上述矛盾给出合理解释. 本文所有整数量的符号如不特别说明均与文[2]的相同，并记 $v(u)$ 为整数 u 的比特数. 关于字长、整数的比特数、比特运算次数、初等运算、运算量和时间复杂度等概念及如何用比特运算次数的概念分析大整数算法的时间复杂度的方

* **收稿日期**：1996-09-16
基金项目：国家教委留学回国人员科研启动基金部分资助项目(教外司留[1996]644 号)
作者简介：张振祥(1947-)，男，江苏盐城人，博士，教授.

法参见文[3-10],这些是计算数论(Computational number theory)(计算机科学和数论相交叉的新学科)中的基本概念和基础知识.

2 算法分析

设 $A=(a_{ik}),B=(b_{kj})$是两个 n 阶非负整数方阵(令文[2]中的 $m=l=n$ 是为简化讨论),矩阵元素的最大值都不超过与 n 无关的某个常数 $a\geqslant 1$. 现在来分析文[2]求 A,B 之积的改进算法(策略一).

由算法第一步,$x=na^2+1>n$.

算法第二步用 Horner 方法计算 $b_k=\sum_{i=1}^{n} b_{ki}x^{i-1}=(\cdots((b_{kn}x+b_{k,n-1})x+b_{k,n-2})x+\cdots+b_{k2})x+b_{k1}$, $k=1,2,\cdots,n$. 所谓 Horner 方法就是依次计算 $H_{k1}=b_{kn}\cdot x+b_{k,n-1},H_{ki}=H_{k,i-1}\cdot x+b_{k,n-i},i=2,3,\cdots,n-1$. 于是 $b_k=H_{k,n-1}$. 因 $x>n$,有 $H_{ki}>x^i>n^i$. 所以 $v(H_{ki})\geqslant v(n^i)>i\log_2 n$.

易见不管用何种快速算法,两个 m 比特整数相乘需不少于 m 次比特运算,一个 m 比特整数和一个 s 比特整数相乘需不少于 $\max(m,s)$次比特运算. 于是计算积 $H_{ki}\cdot x$ 需不少于 $i\log_2 n$ 次比特运算. 计算每个 b_k 则需不少于 $\sum_{i=2}^{n-1} i\log_2 n=\frac{(n+1)(n-2)}{2}\log_2 n=O(n^2\log n)$次比特运算.

这样计算所有 $b_k,k=1,2,\cdots,n$ 需不少于 $O(n^3\log n)$次比特运算. 于是不对其它步骤的运算量作估算就知道[2]中算法的时间复杂度不低于 $O(n^3\log n)$,即其运算量的阶仍高于常规算法的运算量的阶.

文[2]的改进策略二和改进策略三都要以策略一的技术为基础,就不再讨论了.

3 结果分析图判析

文[2]第五节给出布尔矩阵乘法算例分析,但没有给出具体例子,叫人难以捉摸. 所谓布尔矩阵就是元素为 0 或 1 的矩阵,如果要求布尔矩阵之积还是布尔矩阵,它们的元素间的运算就应在域 Z_2 中进行,这样理解的话,文[2]的算法就是错误的,因为它的第四步要做带余除法,而域中是没有带余除法的(或认为余数都是零). 现在我们只好理解为下例.

令 A,B 都是 n 阶方阵,所有元素都为 1 或 0,于是 $a=b=1$. 再令 $n=100$,则 $x=nab+1=101$,若 A,B 的所有元素都为 1,则 $C=AB$ 的所有元素应都为 100. 对此例施不同算法在 PC 机上实测耗时与文[2]的结果分析图(图 1)可能是吻合的. 此图中,Strassen 算法最耗时,文[2]的算法最快,常规算法居中. 这样似乎与算法分析相矛盾,问题又在哪里呢?

记 $T_1(n),T_2(n)$和 $T_3(n)$分别为用 Strassen 算法、文[2]的算法和常规算法在 PC 微机上用 Turbo Pascal 语言对元素为 0 或 1 的 n 阶方阵求积的实测耗时(秒)函数,则有与 n 无关的常数 k_i(参见文[3]的第一节)使 $T_1(n)\leqslant k_1 n^{\log_2 7},T_2(n)\leqslant k_2 n^3\log n$ 和 $T_3(n)\leqslant k_3 n^3$. 在 Strassen 算法中,由于有许多称为簿记(book-keeping)的工作(辅助运算和中间量的存取等)消耗一定时间,所以常数 k_1 在三者中可能为最大. 常数 k_2 在三者中可能为最小,这是因为文[2]的算法人为地把多个形如 1+2=3 的加法合并(即文[2]所说的"凝聚")成稍大一点的数的加法就会

节省一定的时间.但文[2]的"凝聚"术不能超出机器的字长,它不能降低(事实上还提高了)算法时间复杂度的无穷大之阶,所做的仅仅是(对某类特例)降低了耗时函数大 O-记号中隐含的常数 k 的值而已.因此认为文[1,2]的算法"是 $O(n^2)$时间的","达到理论下界"等都是绝对错误的.

不管 k_i 取何值,只要它们都是与 n 无关的常数,就有$\lim\limits_{n\to\infty}\frac{k_2n^3\log n}{k_3n^3}=\infty$,$\lim\limits_{n\to\infty}\frac{k_3n^3}{k_1n^{\log_2 7}}=\infty$.所以当 n 渐渐增大后(例如大到需 2 或 3 个字节储存),文[2]的结果分析图中的几条曲线就会相交,然后必将是文[2]算法的耗时曲线位于最上方(最耗时),Strassen 算法的耗时曲线位于最下方(最快),并且再也不会相交地延伸出去.

参考文献:

[1] 蒋昌俊,吴哲辉. 矩阵乘法的一个最佳算法 [J]. 科学通报, 1989, **34**(4): 251—254.

[2] 蒋昌俊,吴哲辉. "矩阵乘法的一个最佳算法"一文的进一步研究 [J]. 计算物理, 1994, **11**(2): 149—153.

[3] 张振祥,裴定一. 多重精度算术的时间复杂度分析 [J]. 数学的实践与认识, 1994, **3**: 74—76.

[4] 张振祥. 关于矩阵乘法的一个算法的时间复杂度 [J]. 数学研究与评论, 1992, **12**(3): 473—475; 中国实用科技成果大辞典(西南交通大学出版社,1994):20001205; Zbl(德国数学文摘). 796. 68120.

[5] 张振祥. 对"关于矩阵乘法与整数卷积最佳算法运算量的估计"一文的评注 [J]. 计算数学, 1996, **18**(1): 8—11.

[6] 张振祥. 关于整数向量卷积的一个算法的时间复杂度 [J]. 计算数学, 1993, **15**(1): 93—94.

[7] ZHANG Zhen-xiang. *Finding finite B_2-sequences with larger $m-a_m^{1/2}$* [J]. Math. Comp., 1994, **63**: 403—414; Zbl. 801. 11013; MR94k:11109.

[8] 张振祥,曾肯成. 一个 53 位数的分解 [J]. 计算机研究与发展, 1995, **32**(6): 1—4.

[9] 张振祥. Jacobi 和素性测定算法在 PC 上的实现 [J]. 计算机工程与科学, 1996, **18**(2): 23—28.

[10] 张振祥. 多重精度算术软件包的设计与实现 [J]. 计算机研究与发展, 1996, **33**(7): 513—516.

On the Complexity of an Improved Algorithm for Matrix Multiplications

ZHANG Zhen-xiang[1,2]

(1. Dept. of Math., Anhui Normal University, Wuhu 241000;

2. State Key Labotory of Information Security, Graduate School of Uni. of Sci. & Tech. of China, Beijing 100039)

Abstract: The complexity of the conventional algorithm for multiplying two $n\times n$ matrices of non-negative integers is $O(n^3)$. Jiang and Wu [1] gave an "optimal" algorithm with $O(n^2)$ "operations". Later they [2] gave three ways to improve the algorithm. In this paper we conclude that the complexity of the improved algorithm is not yet lower than $O(n^3\log n)$, so its order is still higher than that of the conventional algorithm.

Keywords: matrix multiplications; algorithm analysis; bit operations; time complexity; computational number theory.