讲 座

量子信息讲座续讲

第一讲 量子计算中的因子分解*

张镇九 张昭理

(华中师范大学物理系 相对论研究中心 武汉 430070)

摘 要 因子分解对所有的现行计算机而言是难解的.这是现在通用的公共加密系统的基础.文章介绍了在量子计算机上进行的 Shor 量子算法,即利用量子态的相干叠加和纠缠特性以及量子逻辑门实现量子计算的方法;并着重从理论原理和实验实现这两方面说明利用余因子函数和离散傅里叶变换使这种量子算法为因子分解是有效的.

关键词 量子计算,量子算法,因子分解,余因子函数,离散傅里叶变换

FACTORIZATION IN QUANTUM COMPUTATION

ZHANG Zhen-Jiu ZHANG Zhao-Li

(*Center for Relativity Studies, Department of Physics, Central China Normal University, Wuhan* 430070)

Abstract Factorization is a non-polynomial problem for the present computers. It is the basis of the public cryptography. In this paper, we discuss the Shor's quantum algorithm on a quantum computer, which is the algorithm using the coherence and entanglement of quantum states on which the quantum logic gates act. We then emphasize from the theoretical principle and the experimental realization the efficiency of the quantum factorization by using the remainder function and discrete Fourier transform.

Key words quantum computation, quantum algorithm, factorization, remainder function, discrete Fourier transform

1 引言

密码现在是政府、银行、公司和私人保护其信息交换的重要手段.由于计算机因特网、传递数字签名(它可被证认而不可被复制)、电子商务和数字现金的推广,保密系统的安全性日益重要.现在所用的计算机网络的公共加密系统,多是以大数因子分解的困难为基础.将两个 30 位的素数的乘积进行因子分解,目前世界上运算速度最快的巨型计算机约需要宇宙的寿命这样长的时间.在同样运算速度的量子计算机上,上述问题 10^{-8}s 可解决.可见现行的公共加密系统在量子计算机面前无任何秘密可言,而以全新的量子概念为基础的量子加密系统却有很好的安全性.这也就是为什么美、英、俄、日、德以及我国都很重视量子信息、量子计算和量子通信方面的研究的重要原因之一(其他原因如计算速度高、能模拟量子系统、模拟核试验等).

现行的计算机,对正整数的因子分解是难解问题,即所需要的计算时间随要计算的数的位数的增加以指数方式增长.这种状况在经典计算范围内不可能从本质上解决.现在经典计算机上使用的是 1977 年 Rivest, Shamir 和 Adelman 三人所发明的 RSA 公共加密系统,它利用两个大素数的乘积难以分解来加密.

在量子计算机上进行的 Shor 量子算法对因子分解是有效的,即对正整数的因子分解所需要的计算时间随要计算的数的位数的增加以多项式方式增长.当正整数的位数很大时,以位数的指数方式增长和以位数的多项式方式增长有巨大的差别.这一方面说明量子计算的一个巨大优越性;另一方面,这将

* 国家自然科学基金资助项目
2000 - 02 - 22 收到初稿,2000 - 04 - 26 修回

从根本上破坏所有现行的计算机上使用的公共安全加密系统的安全性. 本文着重从理论原理和实验实现这两方面说明 Shor 量子算法中的因子分解[1—9]的有效性,显示量子计算的巨大优越性. 尽管量子算法可提供新的量子公共安全加密系统,但是不在本文讨论之列.

量子算法中的因子分解的关键是:将若干个正整数因子的乘积分解为各因子等效于求余因子函数的周期;量子计算中的输入态和输出态是处于量子纠缠之中,对余因子函数的输出态进行测量并不能得到它的周期,但是可利用分立傅里叶变换,不测量输出态,而测量输入态,以求得周期,于是实现因子分解,而所需要的时间只随位数的多项式方式增长.

量子计算机是以量子物理作为信息处理的理论基础的、正在研究中的新一代计算机. 迄今为止,现行的各种不同类型的计算机,都是以经典物理学为信息处理的理论基础(尽管某些器件,如半导体器件,与量子物理相关),为与量子计算机相区别而称为经典计算机.

适用于经典计算机上运行的算法称为经典算法. 适用于量子计算机上运行的算法称为量子算法. 模拟量子系统对于经典算法是属于难解问题,对于量子算法却是有效的;因子分解对于经典算法属于难解问题,对于量子算法却是有效的. 但是,量子算法不能做经典算法不能做的问题.

量子计算研究实现量子态的相干叠加和纠缠并对其进行有效处理、传输和测量的方法.

2 经典计算中的因子分解

首先说明因子分解在公共安全加密系统的作用,然后说明如何在经典计算机上进行因子分解.

在计算机上做乘法比较简单,如:127 ×129 = ? 很快可以算出. 反过来,如:? ×? = 29083,求两个因子,就不那么简单. 若两个未知的素数 A 和 B 为未知,它们的积 N 为已知,由已知的 N 去求这两个未知的素数因子 A 和 B,这就是所谓的因子分解问题. N 越大,解起来越困难.

现在最著名、最成功的 RSA 公钥系统是根据数论的研究成果发展出来的. 它使用三个不对称的钥匙,其中两个是公钥,另一个是私钥. 任何人都可以用公开的公钥加密,但需用保密的私钥才可解密.

例如,张三要接收李四发来的加密的信息,张三随意选取两个大的素数 p 和 q,还需要选取两个大数 d 和 e,使 $(de-1)$ 可被 $(p-1)(q-1)$ 除尽. 张三将 p 和 q 的乘积 N(不是 p,也不是 q)和 e 这两个数作为公钥公布,将 d 作为私钥保存. 李四将要发送给张三的信息用数 m 表示,利用张三的公钥 N 和 e,将 m 编为密码 $c = m^e \bmod N$(意思是:c 是 m^e 被 N 除的余数),发给张三. 张三收到 c,利用私钥 d 解密,以得到 $m = c^d \bmod N$. 下面用较小的数为例加以说明. 设张三选取 $N = 15$ $(p = 3, q = 5)$, $e = 3$, $d = 3$,它们满足:$(de-1) = 8$ 可被 $(p-1)(q-1) = 8$ 除尽. 张三公布公钥 $N = 15$ 和 $e = 3$. 设李四要送给张三的信息是 $m = 7$,换算为密码 $c = 7^3 \bmod 15 = 13$. 李四发出 13,张三收到 13. 张三利用自己的私钥 d 解密,以得到 $m = 13^3 \bmod 15 = 7$. 这就是李四向张三传递的秘密信息. 类似地,李四也可以公布两个公钥,保存自己的私钥. 这样,张三和李四之间可以互通密信,不知私钥者无法解读. 要破密,就要能求得私钥. 事实上,只要能将 N 的素数因子 p 和 q 求得,由 $(de-1)$ 可被 $(p-1)(q-1)$ 除尽,可得私钥 d. 在本例中,$N = 15$, $e = 3$,由 $(3d-1)$ 可被 $(3-1)(5-1) = 8$ 除尽,得到密钥 $d = 3$.

因此,N 越难因子分解,私钥就越安全. 实际的公钥系统用的是很大的数 N,很难将其因子分解. 1996 年美国允许出口的公钥系统不能多于 512 位,私钥系统不能多于 56 位. 1977 年 RSA 三人中的 Rivest 提出一个 129 位的数 N:

114381625757888867669235779976146612010218296721242362562561842935706935245733897830597123563958705058989075147599290026879543541,

这就是所谓RSA - 129 问题. 直到 1994 年才得到分解. RSA - 130 问题,1996 得到分解. 1997 年伦敦股票交易所用 CREST 系统是用的 155 位的 RSA 技术. 目前最快的计算机也不能求出其因子[13].

在经典计算机上进行因子分解是逐次用 2,3,4,5,…去除 N,以找到能除尽 N 而没有余数的那些素数. 输入的信息量的大小以二进制的位来量度. 如果有一个正整数 N,则可以找到另一个正整数 L,设它是大于 $\log_2 N$ 的最小正整数,我们称 L 为 N 的在二进制中的位数. 例如,若 $N = 1023$,则 $L = 10$. 若 N 为两个素数的积,则找出其中一个因子所需进行除法运算的次数最大为小于 $N^{1/2}$ 的最大正整数,即小于 $2^{L/2}$ 的最大正整数. 如果用十进制的语言说,我们可以找到正整数 K,设它是从大于的方向最接近于 $\log_{10} N$ 的正整数,我们称 K 为 N 的在十进制中的位数. L 约为 K 的三倍. 例如,若 $N = 1023$,则

$K=3$. 因此,不论在二进制中或在十进制中,用这种算法进行运算的次数以 $2^{L/2}$ 或 $10^{L/6}$ (即都以 L 为指数)的方式增长.

当采用某种算法计算某类问题,其所需的计算次数的增长不快于输入量的大小的多项式函数的增长,则称该种算法对该类问题是有效的,或称该算法对该类问题为有效算法. 反之,当采用某种算法计算某类问题时,其所需的计算次数的增长与输入量的大小按指数函数增长,则称该种算法对该类问题是难解的. 上述经典算法对因子分解是难解的.

以目前世界上运算速度最快的经典巨型计算机(12 万亿次/秒)为例,设二进制的位数 $L=200$, $2^{L/2}=2^{100}\approx 10^{60/2}$,相当于十进制为 60 位的数,作 10^{30} 次运算需 10^{17}s,约为宇宙的寿命(约为 10^{17}s). 考虑到其他因数,量子计算机要达到 $L=500$ 才能显示其优越性.

随后,我们将说明,在量子计算机上采用 Shor 量子算法进行因子分解是有效的. 为了说明如何在量子计算机上采用 Shor 量子算法进行因子分解,下面首先介绍作为量子计算基础的量子位与量子存储器.

3 量子位与量子存储器

在量子物理学中,一个原子中的电子可处于基态或激发态. 一个光子可处于两种偏振态. 我们用 $|0\rangle$ 表示基态或一个偏振态,用 $|1\rangle$ 表示激发态或另一个偏振态. 其中符号 $|\ \rangle$ 表示量子态. 一般地说,类似这样的两个量子态就组成量子两态系统(用 $|0\rangle$ 和 $|1\rangle$ 表示),它可处于它们的叠加态 $|\psi\rangle$,表示为 $|\psi\rangle=a|0\rangle+b|1\rangle$,其中 a 和 b 可为任意两个复数. a 和 b 的取值只受 $|a|^2+|b|^2=1$ 的限制. 因此可有无穷多对 a 和 b 的取值,叠加出的态也就可有无穷多个. 测量 $|\psi\rangle$ 态,得到 $|0\rangle$ 的几率为 $|a|^2$,得到 $|1\rangle$ 的几率为 $|b|^2$.

经典计算机中电流经过二极管的“通”或“断”两种状态作为一个位,这个位的取值可为 0 或 1,对应于“通”或“断”,作为二进制计算的基础.

量子计算机中,一个粒子(如以适当的频率的激光激发处于基态的原子,或通过某种晶体的光子)处于量子两态系统的叠加态,就形成一个量子位. 量子位对应于量子态 $|0\rangle$ 或 $|1\rangle$,作为量子计算机二进制计算的基础. 几个量子位的表示方法也与通常的表示类似. 例如,6 的二进制表示是 110,量子位表示就是 $|110\rangle$,也可以简单地写为 $|6\rangle$. 又例如 $|8\rangle=|1000\rangle$, $|15\rangle=|1111\rangle$.

量子计算机中与经典存储器对应的是量子存储器. 量子计算机的不同之处是需要两个量子存储器,即输入存储器和输出存储器. 而且两个存储器中的量子态处于一种特殊的量子相关联的状态,称为量子纠缠态.

下面我们要说明如何通过量子逻辑门来转换量子态和如何利用量子并行计算来进行计算,以得到输出态.

4 量子逻辑门与量子并行计算

经典计算机中的信息处理是用逻辑门来进行. 量子计算机中的信息处理则是用量子逻辑门或简称量子门来进行,它是一种变换,用算子表示,它将一个态演化为另一个态.

例如,4 位的量子存储器初始都处于 $|0000\rangle$ 态,对每一个位实行量子逻辑门 U 的演化,则可成为

$$|\Psi\rangle=(1/4)(|1111\rangle+|1110\rangle+|1101\rangle+|1011\rangle+|0111\rangle+|1100\rangle+|1010\rangle+|1001\rangle+|0110\rangle+|0101\rangle+|0011\rangle+|1000\rangle+|0100\rangle+|0010\rangle+|0001\rangle+|0000\rangle),$$

即得到 16 个项. 这说明,若将输入存储器制备为若干个数的相干叠加态,接着进行线性运算,则计算的每一步将同时对叠加态中的数同时进行,这就是量子并行计算. 这种量子并行计算与将若干台计算机联起来进行并行计算的本质不同之处是,量子并行计算是对一台计算机的存储器中的各个量子位自动同时进行.

虽然,对输出态进行量子测量所能提供的信息是很有限的,只可提供整个输出态的联合特性(如函数 f 的周期性),而且周期函数的叠加出现的相干效应将大大简化运算. 由于这些特性,我们利用带周期性的余因子函数去进行量子因子分解. 将求数 N 的因子问题化为求余因子函数的周期的问题.

5 余因子函数

一个正整数 N 对正整数 a 的余因子函数 f 是 a 的 x 次方被 N 除的余数:

$$f_{a,N}(x)=a^x \bmod N,$$

其中 a 为与 N 互素的可随机选定的数, $a<N$, x 取零和正整数的数列,即 $x=0,1,2,3,4,\cdots$,设该函数

的周期为 r,则 $f(x)=f(x+r)$. 例如取 $N=15$, a 可取 2,4,7,8,11,13,14,它们符合与 N 互素的要求;在这些可能取的 7 个 a 值中,设进一步取 $a=7$,并按要求取 $x=0,1,2,3,4,5,6,7,8,\cdots$: 相应地, $a(=7)$ 的 x 次方分别为 1,7,49,343,2401,16807,…,它们被 $N(=15)$ 除的余数 $f=1,7,4,13,1,7,4,13,1,7,\cdots$;它们是 4 个数 1,7,4,13 的重复排列,即每隔 4 个数就出现相同的数,于是 f 的周期 $r=4$.

6 离散傅里叶变换

设两个存储器的总初始态为$|0,0\rangle$(这是简化的写法,其中按两个存储器的态依次排列). 接着,通过前述的逻辑门将要计算的数放入输入存储器各个量子位,输入存储器处于各个量子位的状态的叠加态:

$$(1/2^L)\sum|x\rangle,$$

前面的系数是归一化系数. 求和对 x 的所有从 0 到 $2^{2L}-1$ 的整数进行. 再按照前述的函数的量子并行计算,将余因子函数作用于输入存储器,将其结果放在输出存储器,将输入存储器和输出存储器依次排列表示出来,两个存储器的总态成为

$$(1/2^L)\sum|x,f_{a,N}(x)\rangle,$$

式中前面的系数是归一化系数. 求和对 x 的所有从 0 到 $2^{2L}-1$ 的整数进行. 前述的例子,取 $N=15$, $a=7$,则输入存储器的状态依次设置为与 0,1,2,3,4,5,6,7,8,…相对应的量子态,输出存储器的状态按照余因子函数值依次设置为与 1,7,4,13,1,7,4,13,1,…相对应的量子态. 其实,测量前,它们处于叠加态,我们看不见具体的态. 为了形象地说明问题,假想地将两个存储器的状态表示如下:

$$(1/64)(|0\rangle|1\rangle+|1\rangle|7\rangle+|2\rangle|4\rangle+|3\rangle|13\rangle+|4\rangle|1\rangle+|5\rangle|7\rangle+|6\rangle|4\rangle+|7\rangle|13\rangle+|8\rangle|1\rangle+|9\rangle|7\rangle+|10\rangle|4\rangle+|11\rangle|13\rangle+|12\rangle|1\rangle+\cdots),$$

前面的系数是归一化系数. 每一个项都是两个存储器的状态的依次排列. 当对输出存储器进行一次测量,可以得到$|1\rangle,|7\rangle,|4\rangle,|13\rangle$这 4 个态中的一个. 设测得的输出存储器的状态是$|4\rangle$,则输出存储器的其他状态不再存在. 两个存储器的总状态为

$$(1/64)(|2\rangle|4\rangle+|6\rangle|4\rangle+|10\rangle|4\rangle+\cdots)=(1/64)(|2\rangle+|6\rangle+|10\rangle+|14\rangle+|18\rangle+\cdots)|4\rangle.$$

现在有重要意义的是在上述测量之后(括号中的)输入存储器的状态:很显然,后一个态是前一个态加 4,即处于以 4 为周期的各个态的平权叠加,而输出存储器只处于一个确定的状态$|4\rangle$.

为了测量输入存储器的态以得出周期,还需要进行离散傅里叶变换. 离散傅里叶变换是

$$|x\rangle\Rightarrow(1/2^{L/2})\sum|k\rangle\exp(2\pi\mathrm{i}kx/2^L),$$
$$(k=0,1,2,3\cdots),$$

其中求和对 k 进行. 与通常的傅里叶变换不同的是其中 k 只能取分立的值. 这一变换是依次利用单量子位算子 $A(j)$ 和两量子位算子 $B(j,k)$ 的量子逻辑门来实现的. 以作用于态$|2\rangle$为例:

$$|2\rangle\Rightarrow A(0)B(0,1)B(0,2)A(1)B(1,2)A(2)|2\rangle.$$

对于 $L=L$ 的一般情况, L 位的离散傅里叶变换需要作 L 次 $A(j)$ 运算, $L(L-1)/2$ 次 $B(j,k)$ 运算,即共需要作 $L(L+1)/2$ 次量子逻辑门操作. 因此,离散傅里叶变换所需要的计算次数是位数 L 的多项式形式 $L(L+1)/2$. 因此,量子因子分解是有效运算.

现在我们将经典因子分解与量子因子分解作一个简略的比较:因为 $L(L+1)/2\leq L^2$,当 $L>5$,则 $2^L>L^2$. 设 $L=200$, $2^{L/2}=2^{100}\approx10^{60/2}$,即对于十进制 60 位的数进行因子分解,现在最快的经典计算机的运算速度约 10 万亿(10^{13})次/s,要作 10^{30}次运算需 10^{17}s,约为宇宙的寿命(约为 10^{17}s). 在量子计算机上采用量子算法,需要作的运算次数约为 $L^2\approx4\times10^4$,同样的运算速度,10^{-8}s 可完成.

7 量子因子分解的实验实现

我们将讨论的重点放在周期的测量. 作为例子,设输入存储器的量子位数 $L=3$. 输入存储器的状态在二进制中的表示为

$$(1/\sqrt{8})(|000\rangle+|001\rangle+|010\rangle+|011\rangle+|100\rangle+|101\rangle+|110\rangle+|111\rangle).$$

输入存储器和输出存储器的状态$|x,f(x)\rangle$,为

$$|\Psi\rangle=(1/\sqrt{8})\{|0,f(0)\rangle+|1,f(1)\rangle+|2,f(2)\rangle+|3,f(3)\rangle+|4,f(4)\rangle+|5,f(5)\rangle+|6,f(6)\rangle+|7,f(7)\rangle\}.$$

对输入存储器的状态作分立傅里叶变换,输入存储器和输出存储器的状态成为 64 项之和. 因为因子分解的余因子函数是周期函数,用 r 表示函数 $f(x)$ 的周期,并进一步设 r 为偶数.

为了简化所考虑的例子的讨论,设 $r=2$(由后面的测量得出),则 $f(0)=f(2)=f(4)=f(6)$, $f(1)=f(3)=f(5)=f(7)$,上述 64 项之和成为 58

项之和,只有 16 个态. 其中 54 项因为相位因子的作用而完全互相抵消,最后得到下列 4 项:

$$|\Psi\rangle = (1/2)\{|0, f(0)\rangle + |0, f(1)\rangle + |4, f(0)\rangle + e^{i\pi}|4, f(1)\rangle\}.$$

现在看出,由于采用具有周期性的余因子函数,再加上量子态的相干性(相长或相消),使这一步的最后结果得到极大的简化. 在本例中,测量输入存储器的状态,将以 1/2 的几率得到$|0\rangle$或者$|4\rangle$,周期 $r = 2$.

知道 f 的周期 r 后,若 r 为偶数(因为取不同的因子 a,将得到不同的周期,如果一次得到的不是偶数,后面步骤无法进行,就应另取一个 a 再试,直到得到的 r 为偶数)且

$$a^{r/2} \bmod N \neq \pm 1,$$

则可求得两个数 A 和 B:

$$A = a^{r/2} + 1, \quad B = a^{r/2} - 1.$$

再分别求 (A, N) 和 (B, N) 的最大公约数 C 和 D,它们就是 N 的素因子,即 $N = C \times D$. 如前面的例子,取 $N = 15$, $a = 7$, $r = 4$,则 $A = 50$, $B = 48$;而 $(50, 15)$ 的最大公约数 $C = 5$, $(48, 15)$ 的最大公约数 $D = 3$,即 15 的两个因子为 3 和 5.

8 结语

最后,值得指出,虽然量子计算中的因子分解是显示量子计算比经典计算优越得多的重要例子,而且在量子计算的实验实现方面,1999 年 4 月日本的研究人员 Hakamura 等[8]宣布量子计算机研究取得重大进展,他们在超导体固体电子器件创造出量子位并能对其施行电的控制. 随后,Gottesman 和 Chuang[9]提出可用实验上已实现了的远距隐形传态(teleportation)和单量子位操作实现通用的量子计算的方案. 最近,Sackett[10]等实现 4 个粒子的量子纠缠,而且其方法在原则上可推广到多个粒子. 2000 年 4 月,Cirac[11]等论证了在微离子阱中可大尺度化的量子计算机方案. 但是,要使量子计算机成为可实用的计算机,还有许多工作要做[12—14].

参 考 文 献

[1] Shor P W. Algorithms for quantum computation: discrete logarithms and factoring, In: Goldwasser S, Fe S eds. Proceedings of the 35th Annual Symposium on the Foundations of Computer Science. Los Alamitos: IEEE Computer Society Press, 1994. 124—134

[2] 张镇九. 量子计算中的因子分解. 见:香山科学会议第 98 次讨论会"量子通讯与量子计算"报告汇编. 北京, 1998. 47—50[ZHANG Zhen-Jiu. In: Proceedings of the 98th XiangShan Science Conference on Quantum Communications and Computation. Beijing, 1998. 47—50 (in Chinese)]

[3] Berman G P, Doolen G D, Mainieri R *et al*. Introduction to Quantum Computers. Singapore: World Scientific, 1998. 1—187

[4] Williams C P, Clearwater S H. Explorations in Quantum Computing. New York: Springer-Verlag, 1998. 133—142

[5] 张镇九. 计算机工程, 1999, 25: 1[ZHANG Zhen-Jiu. Computer Engineering, 1999, 25: 1 (in Chinese)]

[6] Williams C P ed. Quantum Computing and Quantum Communications. Berlin: Springer, 1999. 1—306

[7] Hey J G ed. Feynman and Computation-Exploring the Limits of Computers. Massachusetts: Perseus Books, 1999. 1—303

[8] Hakamura Y, Pashkin Y A, Tsai J S. Nature, 1999, 398: 786

[9] Gottesman D, Chuang I L. Nature, 1999, 402: 390

[10] Sackett C A, Kleipinski D, King B E *et al*. Nature, 2000, 404: 256

[11] Cirac J I, Zoller P. Nature, 2000, 404: 579

[12] Bennett C H, DiVincenzo D P. Nature, 2000, 404: 247

[13] Brown J. Minds, Machines, and the Multiverse——The Quest for the Quantum Computer. New York: Simon & Schuster, 2000. 1—396

[14] Zhang Z-J (张镇九), Zhang Z-L (张昭理). Nonlocality, entanglement and quantum computers. In: Bergmann P G *et al* eds. Classical and Quantum Nonlocality. Singapore: World Scientific, 2000. 250—281