【北京大学《量子信息物理原理》课程讲稿】（XXI）

量子算法概论

一，量子算法引言

1，经典的计算复杂性理论。

2，量子算法基本特征

二，*Deutsch* 量子算法

1，*Deutsch* 问题

2，*Deutsch* 量子算法步骤

三，量子离散 *Fourier* 变换 DFT_q

1，离散 *Fourier* 变换 DFT_q 定义

2，算法的实施

四，量子 *Shor* 算法

1，任务

2，*Shor* 量子算法步骤简单概括

3，关键步骤是第一步求周期 r

4，量子 *Shor* 算法两点注记

五，量子 *Grover* 算法——“量子摇晃”或量子搜寻算法

1，*Grover* 算法——遍历搜寻问题的量子算法

2，对 *Grover* 算法具体操作说明

3，（23）式证明

4，*Grover* 算法物理实现

※　　※　　※

一，量子算法概论

1，经典计算复杂性理论

在数字计算中，某些计算是快的（如乘法）

$$127\times229=?$$

某些计算则是慢的（因子分解）

$$?\times?=29083$$

引入计算复杂性理论的两个术语：

输入规模 $L=Log_2N$——输入数 N 的二进制码的位数，即为存储它所需 *qubit* 数。

某算法Ω的复杂性 $f_\Omega(L)$——该算法的效果，比如执行该算法所花费的时间 T 或运行步骤数 n。

【定义】如果一种算法Ω的复杂性 $f_\Omega(L)$随输入规模 L 增加以不快于多项式 $Poly$（L）的速度增加，即，如果运算步骤 n 满足，

$$n\le Poly(\log_2 N)$$

这里 $Poly(x)$——x 的任一多项式，就称此算法Ω为快算法或有效算法。否则称为慢算法或无效率算法。

【例 1】大数 N 的质数因式分解

已知某大数是两个质数相乘的结果，求这两个质数因子。若试用 $1\to\sqrt{N}$ 去除，这一算法所需步骤的数量级为 $\sqrt{N}=2^{\frac{1}{2}\log_2 N}$。这是 Log_2N 的指数函数，不是任何有限阶多项式，因而这一算法不是有效算法。

【例 2】海量元素集合中的遍历搜寻

在元素总数为 N 的不同而又随机排列集合中，寻找某一（或某些）所要的元素。比如，已知某种特征或数据去找相应的人

名；或者，只知道北京某人的名字，要在北京市电话簿中找他的电话号码，等等。这类遍历搜寻算法的总步骤也不能表示为 Log_2N 的有限阶多项式，所以也不是一种有效算法。

实际上，还有许多著名计算问题，按经典计算复杂性理论，都不存在快算法。例如：寻求标准 *Boolean* 方程解——适定性问题；3 维匹配问题；顶点复盖问题；*Hamilton* 圈问题；剖分问题；旅行售货员问题；中国邮递员问题；量子系统的经典模拟问题（不可能——*Feynman*）等等。

2， 量子算法的基本特征

众所周知，经典算法理论本身和量子力学毫无关系，也完全不依靠物理学。但现在，量子算法利用量子力学许多基本特性，如相干叠加性、并行性、纠缠性、测量坍缩等等，这些纯物理性质为计算效率的提高带来极大帮助，形成一种崭新的计算模式——量子算法。有些问题，依据经典计算复杂性理论，是不存在有效算法的，但在量子算法的框架里却找到了有效算法。本来，物理学和数学发展的历史从来是物理学利用和依靠数学，现在则是量子力学的物理原理第一次帮助数学去突破数学理论原有的限制：通过量子计算，物理学在真正意义上帮助、发展和改进了计算数学。由此，经典计算复杂性理论需要作重大修改，以便容纳这种崭新的量子计算理论。

一般地说，量子算法有两个存储器 *A* 和 *B*，先将 *A* 的各个 *qubit* 转 $\pi/2$，得到存储器的计算初态

$$\underset{A\quad B}{|0\rangle\rangle\left(\otimes|0\rangle\rangle\right)} \Rightarrow \frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}\underset{A\quad B}{|a\rangle\rangle\left(\otimes|0\rangle\rangle\right)} \qquad (1)$$

这时，为实施算法 *f* 的多重量子逻辑门操作组合成一个总的幺

正算符 $U(f)$。它作用于存储器 A 和 B；利用量子算法的并行性，同时对 A 求和式中所有自变数 a 的每一项作用，一次性地算得相应的全部函数值 $f(a)$；接着，利用 U 中的相互作用，迅即存入 B 中各对应的量子态内，造成两个存储器量子态的纠缠：

$$U(f)\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle\rangle|0\rangle\rangle=\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}U(f)|a\rangle\rangle\otimes|0\rangle\rangle=\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle\rangle\otimes|f(a)\rangle\rangle \qquad (2)$$

然后，测量 A（或 B）存储器，造成 A（或 B）的坍缩，带动 B（或 A）的关联坍缩。最后达到相应计算的目的。

二，*Deutsch* 量子算法

1，*Deutsch* 问题

对单个 *qubit* 变换共有四种方式(输入为 x=0,1)：

$$\left.\begin{aligned}f_1(x)&=x\\f_2(x)&=\bar{x}\end{aligned}\right\}\ ;\quad\left.\begin{aligned}f_3(x)&=0\\f_4(x)&=1\end{aligned}\right\}$$

如何用一次计算即可判断一个未知的 $f(x)$ 属于哪一类型？

Deutsch 问题答案：经典算法必须两次；量子算法只需一次。

2，*Deutsch* 量子算法步骤：

i, 计算初态 $|0\rangle\otimes|0\rangle\rightarrow\frac{1}{2}\sum_{x=0}^{1}|x\rangle\otimes(|0\rangle-|1\rangle)$

ii, 对第一个 *qubit* 两个态 $|x\rangle(x=0,1)$ 执行并行计算 $f(x)$，再利用两个 *qubit* 之间的相互作用，将结果存入第二个 *qubit*。形成如下式左边两个 *qubit* 的纠缠态

$$\boxed{\frac{1}{2}\sum_{x=0}^{1}|x\rangle\otimes(|0+f(x)\rangle-|1+f(x)\rangle)=\frac{1}{2}\sum_{x=0}^{1}(-1)^{f(x)}|x\rangle\otimes(|0\rangle-|1\rangle)} \qquad (3)$$

iii, 结果：第一个 *qubit* 的态为

$$\begin{cases} \frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right), \quad \vec{P}=+\vec{e}_x, \quad f(x)\text{is the constant - type} \\ \frac{1}{\sqrt{2}}\left(|0\rangle-|1\rangle\right), \quad \vec{P}=-\vec{e}_x, \quad f(x)\text{is the balance - type} \end{cases} \tag{4}$$

iv, 对第一个 *qubit* 执行 $Y\left(\frac{\pi}{2}\right)$，测 $\vec{P}$。最后即得

$$\begin{cases} |0\rangle \to f(x)\text{常数型} \\ |1\rangle \to f(x)\text{平衡型} \end{cases} \tag{5}$$

三，量子离散 *Fourier* 变换 DFT_q【1】

经典的离散 *Fourier* 变换。它是对于 N 维复向量 $(x_0,\cdots,x_{N-1})$ 的一个线性映射：

$$(x_0,\cdots,x_{N-1})\to(y_0,\cdots,y_{N-1}) \quad : \quad y_k \equiv \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} e^{2\pi i\, jk/N} x_j$$

现转入量子情况。设有 L 个 *qubit*，组成 *Quantum Storage*，记 $q=2^L$。

【定义】离散 *Fourier* 变换 DFT_q 是如下 q 维基矢的幺正变换：

$$\left\{|a\rangle\rangle\right\}\Rightarrow\left\{|c\rangle\rangle\right\}: |c\rangle\rangle \equiv \frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}\exp\left(2\pi i\, ac/q\right)|a\rangle\rangle, \quad (0\le a,c<q) \tag{6a}$$

这里对所有数都有 $\bmod q$。或者一般地，转记作振幅函数变换：

$$f(a)\Rightarrow\tilde{f}(c): \sum_{a=0}^{q-1} f(a)|a\rangle\rangle \Rightarrow \begin{cases} \sum_{c=0}^{q-1}\tilde{f}(c)|c\rangle\rangle \\ \tilde{f}(c)\equiv\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}\exp\left(2\pi i\, ac/q\right)f(a) \end{cases} \tag{6b}$$

【DFT_q 算法实施】

i, 取下面两种变换（$\theta_{jk}=\pi/2^{k-j}$）

$$\begin{cases} A_j=\frac{1}{\sqrt{2}}\left(\sigma_x^{(j)}-\sigma_z^{(j)}\right) & for\ \ j-qubit \\ B_{jk}=e^{i\theta_{jk}\hat{n}_j\hat{n}_k}=1+\hat{n}_j\hat{n}_k\left(e^{i\theta_{jk}}-1\right) & for\ \ j-and\ k-qubits \end{cases} \tag{7a}$$

A_j 只对第 j 个单个 *qubit* 作用，其中 $\sigma_x^{(j)}$ 对 j 位实施位翻转，$-\sigma_z^{(j)}$

定义相位反转：对 j 位的 1 态反号，对 j 位的 0 态不反号。B_{jk} 同时对第 j、第 k 两个 *qubit* 作用：当且仅当这两个 *qubit* 都是 1 时，添一个相因子，对测 $\bar{P}$，其它情况此算符无作用。

$$\therefore \quad \begin{cases} A_j\left|a_j\right\rangle = \dfrac{1}{\sqrt{2}}\sum_{b_j=0}^{1}\exp\left(\pi i a_j b_j\right)\left|b_j\right\rangle \\ B_{jk}\left|b_k\right\rangle\otimes\left|a_j\right\rangle = \exp\left(i\theta_{jk}a_j b_k\right)\left|b_k\right\rangle\otimes\left|a_j\right\rangle \end{cases} \qquad \left(a_j, b_k = 0,1\right) \qquad \textbf{(7b)}$$

ii, 以下面序列作用于输入态

$$\left(A_0 B_{0,L-1}\cdots B_{0,1}\right)\cdots\left(A_j B_{j,L-1}\cdots B_{j,j+1}\right)\cdots\left(A_{L-2}B_{L-2,L-1}\right)A_{L-1}\left|input\right\rangle$$

iii, 将结果态矢内二进制码字顺序全部颠倒，得最后结果

$$\boxed{DFT_q\left|a_{L-1},a_{L-2},\cdots,a_0\right\rangle = \left(A_0 B_{0,L-1}\cdots B_{0,1}\right)\cdots A_{L-1}\left|input\right\rangle\Big|_{change}} \qquad \textbf{(8)}$$

【$\left|3\right\rangle\!\rangle = \left|011\right\rangle$ 算例】

$$A_0 B_{02} B_{01} A_1 B_{12} A_2\left|011\right\rangle = \left(\frac{1}{\sqrt{2}}\right)^3\left\{\left|000\right\rangle - \left|001\right\rangle - e^{i\frac{\pi}{2}}\left|010\right\rangle + e^{i\frac{\pi}{2}}\left|011\right\rangle + e^{i\frac{\pi}{4}}\left|110\right\rangle - e^{i\frac{\pi}{4}}\left|111\right\rangle + e^{i\frac{3\pi}{4}}\left|100\right\rangle - e^{i\frac{3\pi}{4}}\left|101\right\rangle\right.$$

$$\Rightarrow \quad DFT_q\left|011\right\rangle = \frac{1}{\sqrt{8}}\left\{\left|000\right\rangle - \left|100\right\rangle - e^{i\frac{\pi}{2}}\left|010\right\rangle + e^{i\frac{\pi}{2}}\left|110\right\rangle + e^{i\frac{\pi}{4}}\left|011\right\rangle - e^{i\frac{\pi}{4}}\left|111\right\rangle + e^{i\frac{3\pi}{4}}\left|001\right\rangle - e^{i\frac{3\pi}{4}}\left|101\right\rangle\right\}$$

显然，此结果正是 DFT_q 定义的如下结果：

$$DFT_q\left|3\right\rangle\!\rangle = \frac{1}{\sqrt{8}}\left\{\left|0\right\rangle\!\rangle + e^{i\frac{3\pi}{4}}\left|1\right\rangle\!\rangle - e^{i\frac{\pi}{2}}\left|2\right\rangle\!\rangle + e^{i\frac{\pi}{4}}\left|3\right\rangle\!\rangle - \left|4\right\rangle\!\rangle - e^{i\frac{3\pi}{4}}\left|5\right\rangle\!\rangle + e^{i\frac{\pi}{2}}\left|6\right\rangle\!\rangle - e^{i\frac{\pi}{4}}\left|7\right\rangle\!\rangle\right\}$$

四，量子 *Shor* 算法【3】

1, 任务： 设有一个很大的奇数 N 为两个质数 n_1 和 n_2 的乘

积。现在计算任务是：已知 N，往求 n_1 和 n_2。

算法背景：按经典计算复杂性理论，这个问题不存在有效算法，所以是各类加密编码方法最初的理论基础。

2, *Shor* 量子算法步骤简单概括：

a) 随机取 $y<N$ 并与 N 互质(即它们俩最大公约数——*the greatest common divisor*，$gcd(y,N)=1$)。用 *Shor* 量子算法求下面函数 $F_N(a)$ 的周期 r：

$$\boxed{F_N(a)=y^a \bmod N} \qquad (9a)$$

这里 $mod\ N$ 的意思是从数 y^a 中减去 N 的正数倍，只留下余数。

注意，这个余数构成数 a 的周期函数。因为，设 *Phi-function* $\varphi(N)$ 的数值是小于给定正整数 N 并和 N 互质的正整数的个数。利用数论中以函数 $\varphi(N)$ 表示的

【*Euler* 定理】 $y^{\varphi(N)} \equiv 1\ mod\ N$ *if* $gcd(y,N)=1$”。

Euler 定理是说，当 $gcd(y,N)=1$ 时，函数 $\varphi(N)$ 肯定是此余子式的周期，但反之未必。即，余子式的周期并不总是函数 $\varphi(N)$。由于现在有 $gcd(y,N)=1$，按 *Euler* 定理，周期 r 一定存在。即有

$$y^r = 1\ mod\ N \quad \Rightarrow \quad y^{b+r} = y^b\ mod\ N \qquad (9b)$$

b) 若 r 为奇数，返回 *a*)重新取 y；若 r 为偶数，取 $y^{\frac{r}{2}} \equiv x$，由上式就得到

$$\boxed{x^2 = 1\ \ mod\ N \ \Rightarrow\ x^2 - 1 = 0\ \ mod\ N} \qquad (10a)$$

说明此同余式方程左边的数可以被 $N = n_1 n_2$ 所整除。

c) 接着分解(10a)式，确定 x 与所求 n_1 和 n_2 的关系。由于已知 $N = n_1 n_2$，并且 $gcd(n_1, n_2)=1$，按孙子定理，求解方程(10a)中的 x 等价于求解下面的同余式方程组（详细见后）：

$$\left\{\begin{array}{l} x_1-1=0\ mod\ n_1 \\ x_1+1=0\ mod\ n_2 \end{array}\right.;\ or\ \left\{\begin{array}{l} x_2-1=0\ mod\ n_2 \\ x_2+1=0\ mod\ n_1 \end{array}\right. \qquad (10b)$$

d) 方程（10b）说明，$(x-1)$和$(x+1)$可以分别为n_1或n_2尽除，也即它们与所求n_1或n_2是整数倍关系。于是，n_1和n_2必定可以由$(x-1)$及$(x+1)$与N的公因子给出，即有

$$\begin{array}{l} n_1=gcd\left(x-1,N\right) \\ n_2=gcd\left(x+1,N\right) \end{array} \qquad (11)$$

现在可以采用“*Euclid* 算法”(用“**余数展转相除法**”求最大公约数)得到这两个公因子n_1和n_2。最后经验算$n_1\cdot n_2=N$予以确定。

3, 上面步骤中最关键是第一步求周期r。按 *Shor* 量子算法它被规划为：

a) 将“$y^{a+r}mod\ N=y^a mod\ N$的周期r”问题等价转化为“$y^r=1 mod\ N$”问题。

b) 用量子逻辑门按算法组合而成的幺正变换去计算y^a的余数，存入第二个存储器中

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}\left|a\right\rangle\left|0\right\rangle\xrightarrow[storage\ into\ second]{caculate\ y^a modN}\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}\left|a\right\rangle\left|y^a modN\right\rangle \qquad (12)$$

c) 对第二个存储器进行测量，设它坍缩到某个z值，

$$z=y^l modN \qquad (13)$$

注意对同一个余数值z，会有多个对应的l值，表明求和式（12）将会塌缩到许多项之和。详细些说是

$$y^{jr+l}\ mod\ N=y^l\ mod\ N=z \qquad (14)$$

即对同一个z，有以下各值（A是在$q-l$区间内最多能容纳r的个数$A<\left[(q-l)/r\right]$）：

$$a=l,l+r,l+2r,\cdots,l+Ar \qquad (15)$$

在对第二个存储器测量时，第一个存储器量子态关联坍缩成为

$$\boxed{\left|\varphi_l\right\rangle\rangle=\frac{1}{\sqrt{A+1}}\sum_{j=0}^{A}\left|jr+l\right\rangle\rangle} \tag{16}$$

这个态是以 r 为周期的一组数态的叠加。虽然存在两个端部的影响，但由于 N 巨大，求和项数很多，左边总和态按照右边各个成分态数字 $jr+l$ 的变化来看，还是非常近似于周期函数。

d) 为了找出这个周期 r 值，对第一个存储器作快速 *Fourier* 变换 DFT_q

$$\boxed{\begin{aligned}&DFT_q\left|\varphi_l\right\rangle\rangle=\sum_{c=0}^{q-1}\tilde{f}(c)\left|c\right\rangle\rangle\\&\tilde{f}(c)=\frac{\sqrt{r}}{q}\sum_{j=0}^{[q/r-1]}\exp\left[2\pi i\left(jr+l\right)c/q\right].\end{aligned}} \tag{17}$$

如同平常 *Fourier* 变换那样，由于 $|\varphi_l\rangle\rangle$ 近似于周期函数，经受快速 *Fourier* 变换的结果， **(17)** 式右边求和中的振幅分布呈现明显的集中。这为下一步测量塌缩提供了较好的成功概率。

e) 对第一个存储器作测量

测量以 $Prob(c)=\left|\tilde{f}(c)\right|^2$ 概率坍缩到 $\left|c\right\rangle\rangle$ 态，同时也得到了 c 值。由于 q 和 q/r 都十分大，c 值通常应当落在位相数值较小区间内

$$\boxed{-r/2\le rc\bmod q\le r/2} \tag{18}$$

否则 $\tilde{f}(c)$ 中求和式各项相因子的相位差异较大，更由于求和项数太多，大量相位差异较大的相因子将会互相抵消。于是向这种 $\tilde{f}(c)$ 坍缩的概率将很小。**(18)**式说明，乘积 rc 的数值大体是 q 值的整数倍，相差数值不大于 r 值。就是说，存在正整数 $c',(0\le c'\le r-1)$，使得 rc 在减去 c' 倍的 q 之后，差值的绝对值将落在 $\frac{r}{2}$ 之内，即应当有

$$\boxed{\left|rc-c'q\right|\le\frac{r}{2}\rightarrow\left|\frac{c}{q}-\frac{c'}{r}\right|\le\frac{1}{2q}} \tag{19}$$

这里$c,q\left(q\ge N^2\right)$为已知，r,c'是未知比值数，但所求比值是两个正整数之比，总是有理数。按照连分数的一个定理，可以将$\frac{c'}{r}$表示为在$\frac{c}{q}$连分数基础上的一个有限阶连分数【2】。

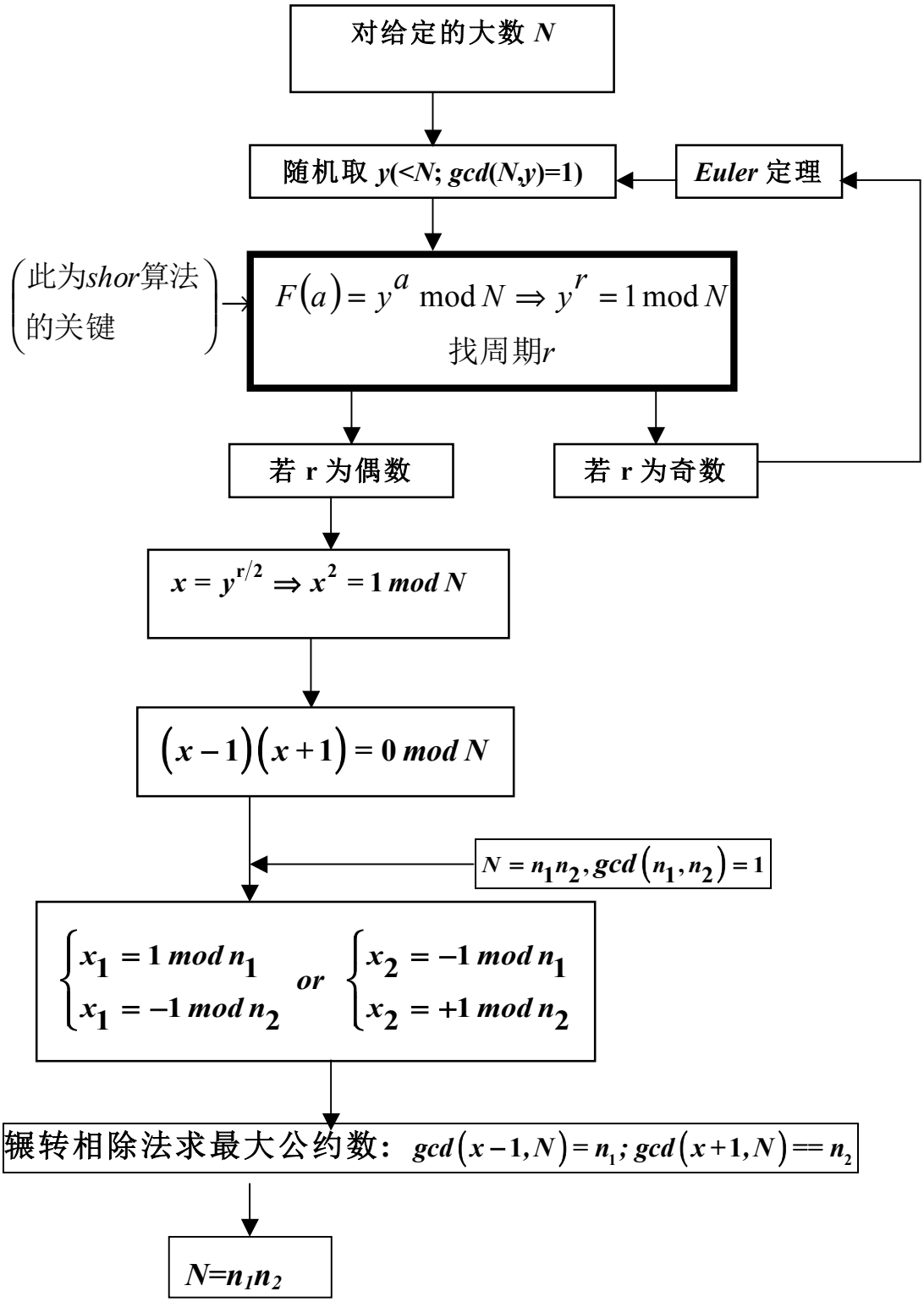
于是，若c'和r互质，即$gcd\left(c',r\right)=1$，由连分数定理所得比值$\frac{c'}{r}$可直接获得r值。此时将所得r值代入$y^r=1\,mod\,N$式中很容易检验其正确性。如不互质，就是说它俩有公因子，由于$0\le c'\le r-1$，此时得出的r值将不是真正的r值，而是r的一个因子。由验算易知是错的（注意问题本身就是求解答案很难，但检验答案却很容易！）。于是必须返回重新计算。

如同所有的量子算法一样，因为涉及到量子测量和塌缩，*Shor* 算法也是一种概率算法。它适用于求解很难，但检验却很容易的场合。

可以证明上述算法是有效算法。就是说，考虑到 *Shor* 算法各步骤的成功概率，对任给的一个小正数ε，总存在（依赖ε的）关于输入长度log_2N的一个多项式$Poly(log_2N)$数目的步骤，使得 *Shor* 算法在运行这么多步之后，成功给出N的因子n_1和n_2的概率大于$(1-\varepsilon)$【1】。

4，量子 *Shor* 算法的两点注记

a)《*Shor* 量子算法流程图》



b) 孙子定理——《*Chinese remainder theorem*》【5】

这个定理最初为我国古代《孙子算经》所载，是关于求解同余式方程组的。问题是：“今有物不知其数，三三数之剩 2，五

五数之剩 3，七七数之剩 2。问物几何？”

答案: 23。

求解可用程大位所著《算法统宗》中的歌诀。歌诀为：

$$\begin{cases} \text{三人同行七十稀。} & (3-70) \\ \text{五树梅花廿一枝。} & (5-21) \\ \text{七子团圆正半月。} & (7-15) \\ \text{除百零五便得知。} & (\text{mod}\,105) \end{cases}$$

m_i	3	5	7
$M_i = M/m_i$	35	21	15
$N_i = M_i^{-1}$ $(mod\ m_i)$	2	1	1

这里 $M = m_1 m_2 m_3 = 105$， $gcd(m_i, M_i) = 1$， $N_i M_i = 1(mod\ m_i)$。根据 m_i 求得 M_i, N_i 之后，设 a_i 为余数，即得

$$\begin{aligned} x &= (a_1 N_1 M_1 + a_2 N_2 M_2 + a_3 N_3 M_3)(mod\ M) \\ &= (2\times 70 + 3\times 21 + 2\times 15)(mod 105) = 233(mod 105) = 23 \end{aligned}$$

一般情况的同余式组：

$$\left(gcd(m_i m_j) = 1, \forall i, j\right): \begin{cases} x = a_1\ mod\ m_1 \\ \vdots \\ x = a_k\ mod\ m_k \end{cases} \Rightarrow x = \sum_{i=1}^{k} a_i N_i M_i \left(mod\ M\right)$$

c）《*Shor* 算法一个简单的例算说明》

给定 *N*=21　　（ $n_1 n_2 = 3\times 7$ ）

取　　*Y*=11

$F(a) = 11^a \bmod N$:

$F(a+r) = 11^{a+r} \bmod N = 11^a \bmod N = F(a)$, *period* $r(\bmod N)$

$a=$	1	2	3	4	5	6	7	8	9	10	11	12
$11^a\ mod\ 21=$	11	16	8	4	2	1	11	16	8	4	2	1

$$11^6 = 1 \ mod \ 21$$

$$\left\{ \Rightarrow 11^{12} = 1 \ mod \ 21 \left(\because 11^{12} - 1 = \left(11^6 + 1\right)\left(11^6 - 1\right) = \left(11^6 + 1\right) \cdot k \cdot 21 \right.\right.$$
$$\left.\left. = l \cdot 21 = 0 \ mod \ 21 \right) \right\}$$

∴ $r = 6$(注解：此处周期是 6。若 $\varphi(N)$ 的定义是小于 21 并与 21 互质的质数个数，则 $\varphi(21) = 6$。但按 *Euler* 定理，$\varphi(N)$ 是小于 21 并与 21 互质的正整数个数，实际 $\varphi(21) = 12$，非此处所要)。

$$x = y^{\frac{r}{2}} = 11^3 = 1331 \ mod \ 21 = 8$$

$$x \pm 1 = 9, 7$$

$$gcd(9, 21) = 3; \qquad gcd(7, 21) = 7$$

$$\therefore \qquad N(21) = 3 \times 7$$

五，*Grover* 算法——“量子摇晃”或量子搜寻算法【4】

L.K.Grover : “*Quantum Mechanics Helps in Searching for a Needle in a Haystack*”， *Phys. Rev. Lett., Vol. 79, No.2, 1997*

1, *Grover* 算法——遍历搜寻问题的量子算法

遍历搜寻问题的任务是从一个海量元素的无序集合中，找到满足某种要求的元素。要验证给定元素是否满足要求很容易，但反过来查找这些合乎要求的元素则很费事，因为这些元素并没有按要求进行有序的排列，并且数量又很大。在经典算法中，只能按逐个元素试下去，这也正是“遍历搜寻”这一名称的由来。此问题用 *Grover* 算法解决已经不再需要“遍历”了，但人们仍然沿袭着历史上的称呼。显然，在经典算法中，运算步骤 *n* 与被搜寻集合中元素数目 *N* 成正比。若该集合中只有一个元素符合要求，为使搜寻成功率趋于 100%，一般说来步骤数 *n* 要接近于 *N*。而在 *Grover* 算法中，使搜寻成功的运算步骤 *n* 只与 $\sqrt{N}$

成正比。由此看来，与经典算法相比，*Grover* 算法的高效率是一目了然的；而且 N 越大越能显示出 *Grover* 算法的优越性。

2, 下面对 *Grover* 算法的具体操作加以说明

设被查找的集合为 $\{|i\rangle\}=\{|0\rangle,|1\rangle,\cdots|N-1\rangle\}$；从这类问题的实际意义出发，可以假定 $N>>1$。假设所有符合所设条件的元素组成集合 Z。不妨先考虑此集合 Z 中元素是唯一的情况，设此元素为 $|x\rangle$。在开始查找之前要对系统进行初始化（对每一个 *qubit* 的初态进行 *Hadamard* 变换），使之处于 $|\varphi_0\rangle=\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle$ 态上。定义算符 $\hat{C}$

$$\boxed{\hat{C}|i\rangle=\begin{cases}|i\rangle, |i\rangle\neq|x\rangle\\ -|i\rangle, |i\rangle=|x\rangle\end{cases}} \tag{20a}$$

这种算符很容易构造，比如令 $\hat{C}=1-2|x\rangle\langle x|$。它的作用是把符合条件态前面的系数变号；而另一个算符为

$$\hat{D}\equiv 2\hat{P}-\hat{I}=\frac{2}{N}\begin{pmatrix}1&1&\cdots&1\\1&1&\cdots&1\\1&1&\cdots&1\\1&1&\cdots&1\end{pmatrix}-\begin{pmatrix}1&&&0\\&1&&\\&&\ddots&\\0&&&1\end{pmatrix} \tag{20b}$$

这里 $\hat{P}$ 是一个对展开式中所有本征数态的系数进行平均的算符，$\hat{I}$ 是单位算符。现把算符 $\hat{D}$、$\hat{C}$ 反复作用在 $|\varphi_0\rangle$ 上，并称一次作用为一次“迭代”。为了研究 n 次迭代

$$\boxed{|\varphi_n\rangle\equiv\left(\hat{D}\hat{C}\right)^n|\varphi_0\rangle}$$

的表示，先把 $|\varphi_0\rangle$ 写为

$$\boxed{|\varphi_0\rangle=\frac{1}{\sqrt{N}}|x\rangle+\frac{1}{\sqrt{N}}\sum_{\substack{i=0\\(i\neq x)}}^{N-1}|i\rangle\equiv|\alpha\rangle+|\beta\rangle}$$

现将 $\hat{D}$、$\hat{C}$ 分别作用在 $|\alpha\rangle$ 和 $|\beta\rangle$ 上，经计算可得

$$\begin{cases}\hat{D}\hat{C}|\alpha\rangle=\left(1-\dfrac{2}{N}\right)|\alpha\rangle-\dfrac{2}{N}|\beta\rangle\\ \hat{D}\hat{C}|\beta\rangle=\left(2-\dfrac{2}{N}\right)|\alpha\rangle+\left(1-\dfrac{2}{N}\right)|\beta\rangle\end{cases}\qquad(21)$$

这表明$|\varphi_n\rangle\equiv\left(\hat{D}\hat{C}\right)^n|\varphi_0\rangle\equiv a_n|\alpha\rangle+b_n|\beta\rangle$总是$|\alpha\rangle$和$|\beta\rangle$的线性组合。将基矢变换的（21）式转置，转换为系数a_n、b_n的递推关系（$\varepsilon=\dfrac{2}{N}$）：

$$\begin{pmatrix}a_n\\ b_n\end{pmatrix}=\begin{pmatrix}1-\varepsilon, & 2-\varepsilon\\ -\varepsilon, & 1-\varepsilon\end{pmatrix}\begin{pmatrix}a_{n-1}\\ b_{n-1}\end{pmatrix},\ \text{and}\ \begin{cases}a_0=1\\ b_0=1\end{cases}\qquad(22)$$

按$N>>1$近似条件，经过n次迭代，直接计算可知（证明见后）：

$$\begin{cases}a_n\approx\sqrt{N}\sin\left(\dfrac{2n}{\sqrt{N}}\right)\\ b_n\approx\cos\dfrac{2n}{\sqrt{N}}\end{cases}\qquad(23)$$

$$\therefore\qquad|\varphi_n\rangle\approx\sin\left(\frac{2n}{\sqrt{N}}\right)|x\rangle+\frac{\cos\left(\dfrac{2n}{\sqrt{N}}\right)}{\sqrt{N}}\sum_{\substack{i=0\\(i\neq x)}}^{N-1}|i\rangle\qquad(24)$$

若此时对$|\varphi_n\rangle$进行测量，按量子力学的测量公设，它坍缩到$|x\rangle$态的概率为

$$P(n)=\left|\sin\left(\frac{2n}{\sqrt{N}}\right)\right|^2\qquad(25)$$

理想的迭代数次数应能使$P(n)$尽量接近 1。$P(n)$是 n 的周期函数（因为上面的迭代运算都是幺正的，因而是可逆的，此时表现为对迭代次数为周期的）；但我们显然应取满足条件的最小正整数 n 值。故取$n_0=\left[\dfrac{\pi}{4}\sqrt{N}\right]$，其中符号[ ]代表用四舍五入法取整数（不是数学中的取不超过原数的整数）。n 为整数这一限制使

得 $P(n_0)$并非 100%，但由于 $\frac{2n_0}{\sqrt{N}}\in\left[\frac{\pi}{2}-\frac{1}{\sqrt{N}},\frac{\pi}{2}+\frac{1}{\sqrt{N}}\right]$，搜寻失败的概率

$$1-p(n_0)\le\cos^2\left(\frac{\pi}{2}-\frac{1}{\sqrt{N}}\right)=0\left(\frac{1}{N}\right) \qquad (26)$$

在 $N>>1$时，失败的概率可忽略不计。这样，*Grover* 算法的每次迭代中用 $\hat{C}$ 算符将符合条件的态 $|x\rangle$ 系数反向，而且在逐次反向过程中，$|x\rangle$ 在 $|\varphi_n\rangle$ 中所占比例越来越大，终于（经量子测量的波包坍缩遂）“脱颖而出”。人们把这种算法形象地称为“*Grover* 量子摇晃”——“量子抽签”（*Grover's Quantum Shake*）。

3，下面证明（23）式。为了更清楚地看出 a_n、b_n 随 n 变化的趋势，下面求它们的通项公式。由（22）式知：

$$\begin{pmatrix}a_n\\b_n\end{pmatrix}=\begin{pmatrix}1-\varepsilon, & 2-\varepsilon\\ -\varepsilon, & 1-\varepsilon\end{pmatrix}^n\begin{pmatrix}1\\1\end{pmatrix} \qquad (27)$$

往求这个变换矩阵的 n 次幂。为此先求得变换矩阵的本征值，为

$$\lambda_+=1-\varepsilon+\sqrt{\varepsilon(\varepsilon-2)},\qquad \lambda_-=1-\varepsilon-\sqrt{\varepsilon(\varepsilon-2)} \qquad (28)$$

设矩阵 $\begin{pmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix}^{-1}$ 可将它对角化，即设

$$\begin{pmatrix}1-\varepsilon & 2-\varepsilon\\ -\varepsilon & 1-\varepsilon\end{pmatrix}=\begin{pmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix}\begin{pmatrix}\lambda_+ & 0\\ 0 & \lambda_-\end{pmatrix}\begin{pmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix}^{-1}$$

右乘 $\begin{pmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix}$，求得

$$\begin{pmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix}=\begin{pmatrix}\sqrt{\varepsilon-2} & \sqrt{\varepsilon-2}\\ -\sqrt{\varepsilon} & \sqrt{\varepsilon}\end{pmatrix}$$

于是

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}^{-1} = \frac{1}{2}\begin{pmatrix} \dfrac{1}{\sqrt{\varepsilon-2}} & \dfrac{-1}{\sqrt{\varepsilon}} \\ \dfrac{1}{\sqrt{\varepsilon-2}} & \dfrac{1}{\sqrt{\varepsilon}} \end{pmatrix}$$

$$\therefore \quad \begin{pmatrix} 1-\varepsilon & 2-\varepsilon \\ -\varepsilon & 1-\varepsilon \end{pmatrix}^{n} = \begin{pmatrix} \sqrt{\varepsilon-2} & \sqrt{\varepsilon-2} \\ -\sqrt{\varepsilon} & \sqrt{\varepsilon} \end{pmatrix}\begin{pmatrix} \lambda_+^n & 0 \\ 0 & \lambda_-^n \end{pmatrix}\begin{pmatrix} \dfrac{1}{2\sqrt{\varepsilon-2}} & \dfrac{-1}{2\sqrt{\varepsilon}} \\ \dfrac{1}{2\sqrt{\varepsilon-2}} & \dfrac{1}{2\sqrt{\varepsilon}} \end{pmatrix} \tag{29}$$

将其代入 a_n、b_n 表达式，得到

$$\boxed{\begin{pmatrix} a_n \\ b_n \end{pmatrix} = \frac{1}{2}\begin{pmatrix} \lambda_+^n\left(1-\sqrt{\dfrac{\varepsilon-2}{\varepsilon}}\right)+\lambda_-^n\left(1+\sqrt{\dfrac{\varepsilon-2}{\varepsilon}}\right) \\ \lambda_+^n\left(1-\sqrt{\dfrac{\varepsilon}{\varepsilon-2}}\right)+\lambda_-^n\left(1+\sqrt{\dfrac{\varepsilon}{\varepsilon-2}}\right) \end{pmatrix}} \tag{30a}$$

将 $\lambda_\pm$ 代入此式，再利用 N(和n)$>>1$的假定，即近似得到（23）式：

$$\begin{cases} a_n \approx \dfrac{-i\sqrt{N}}{2}\left[\left(1+i\sqrt{\dfrac{\varepsilon}{2}}\right)\left(1+i\sqrt{2\varepsilon}\right)^n-\left(1-i\sqrt{\dfrac{\varepsilon}{2}}\right)\left(1-i\sqrt{2\varepsilon}\right)^n\right] \\ \quad \approx \dfrac{\sqrt{N}}{2i}\left(e^{i\frac{2n+1}{\sqrt{N}}}-e^{-i\frac{2n+1}{\sqrt{N}}}\right)=\sqrt{N}\sin\left(\dfrac{2n+1}{\sqrt{N}}\right)\cong\sqrt{N}\sin\left(\dfrac{2n}{\sqrt{N}}\right) \\ b_n \approx \dfrac{1}{2}\left[\left(1-i\sqrt{\dfrac{\varepsilon}{2}}\right)\left(1+i\sqrt{2\varepsilon}\right)^n-\left(1+i\sqrt{\dfrac{\varepsilon}{2}}\right)\left(1-i\sqrt{2\varepsilon}\right)^n\right] \\ \quad \approx \dfrac{1}{2}\left(e^{i\frac{2n-1}{\sqrt{N}}}+e^{-i\frac{2n-1}{\sqrt{N}}}\right)=\cos\left(\dfrac{2n-1}{\sqrt{N}}\right)\cong\cos\left(\dfrac{2n}{\sqrt{N}}\right) \end{cases} \tag{30b}$$

4，*Grover* 算法的物理实现。

为方便起见，通常取 $N=2^l$，l 为正整数。这样，就可以用 l 个量子位按照二进制编码的规则来表示 $|i\rangle\rangle$。按前面第四节叙述，将 l 个量子位的存储器转入运算初态 $|\varphi_0\rangle\rangle$，

$$|\varphi_0\rangle\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)\otimes\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)\otimes\cdots\otimes\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)$$
$$=\frac{1}{\sqrt{2^l}}\left(|00\cdots0\rangle+|00\cdots1\rangle+\cdots|11\cdots1\rangle\right)=\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\rangle$$

定义 $\hat{C}_x$ 为

$$\hat{C}_x|i\rangle\rangle=\begin{cases}|i\rangle\rangle, & i\notin X\\ -|i\rangle\rangle, & i\in X\end{cases}；其中，\ \hat{C}_0|i\rangle\rangle=\begin{cases}|i\rangle\rangle, & i\neq 0\\ -|i\rangle\rangle, & i=0\end{cases} \tag{31}$$

引入 *Walsh-Hadamard* 变换 $\hat{T}$，

$$\boxed{\hat{T}|i\rangle\rangle=\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\rangle} \tag{32}$$

其中“$i\cdot j$”表示 i 与 j 两个数的二进制表示序列逐位相乘并求和，即，如果 $i=\left(C_{i,l-1}C_{i,l-2}\cdots C_{i,0}\right)$，$j=\left(C_{j,l-1}C_{j,l-2}\cdots C_{j,0}\right)$（各位的 C 值只能取 0 或 1），则 $i\cdot j\equiv\sum_{k=0}^{l-1}C_{ik}C_{jk}$。

【证明】*Grover* 算法中迭代可用算符 $\hat{G}\equiv-\hat{T}\hat{C}_0\hat{T}\hat{C}_x$ 实现【4】。

设 $|\varphi\rangle$ 为 $|0\rangle\rangle,|1\rangle\rangle,\cdots|N-1\rangle\rangle$ 的任一组合 $|\varphi\rangle=\sum_{j=0}^{N-1}a_j|j\rangle\rangle$；这里已经定义 $|\varphi'\rangle\equiv\hat{C}_x|\varphi\rangle$，$a'_k\equiv\begin{cases}a_k, & k\notin X\\ -a_k, & k\in X\end{cases}$。于是有

$$\begin{aligned}\hat{G}|\varphi\rangle&=-\hat{T}\hat{C}_0\hat{T}|\varphi'\rangle=-\hat{T}\hat{C}_0\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\sum_{i=0}^{N-1}(-1)^{i\cdot j}a'_j|i\rangle\rangle\\&=-\hat{T}\frac{1}{\sqrt{N}}\left[-2\left(\sum_{j=0}^{N-1}a'_j|0\rangle\rangle\right)+\sum_{j=0}^{N-1}\sum_{i=0}^{N-1}(-1)^{i\cdot j}a'_j|i\rangle\rangle\right]\\&=\left(\frac{2}{N}\sum_{j=0}^{N-1}a'_j\right)\sum_{k=0}^{N-1}|k\rangle\rangle-\frac{1}{N}\sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\sum_{k=0}^{N-1}(-1)^{i\cdot j}(-1)^{i\cdot k}a'_j|k\rangle\rangle\end{aligned} \tag{33}$$

这时注意有

$$\sum_{i=0}^{N-1}(-1)^{i\cdot j}(-1)^{i\cdot k}=N\delta_{jk} \tag{34}$$

这是因为，当 $j=k$，不难看出此式精确等于 N；当 $j\neq k$，对全体 i 求和时各项的指数 $i\cdot j$ 和 $i\cdot k$（其实是两者之和）的奇偶性决定此项为 -1 或 $+1$。但现在 N 很大，二进制各位对应相乘所得的这两个指数的奇偶性概率近似相等，故对 i 求和最终很小于 N。

$$\therefore \quad \hat{G}|\varphi\rangle = 2\hat{P}|\varphi'\rangle - \sum_{k=0}^{N-1} a'_k |k\rangle\rangle = \left(2\hat{P} - \hat{I}\right)\hat{C}_x|\varphi\rangle = \hat{D}\hat{C}_x|\varphi\rangle \tag{35}$$

以上简略介绍表明，*Grover* 量子摇晃算法的确是解决遍历搜寻问题的一种有力工具。它将搜寻次数与搜寻长度的关系由 $N \to \sqrt{N}$。当 N 较大时，它的优越性体现得尤其明显。例如，用 *Grover* 算法破解通用的 56 位加密标准（*DES*），只需 $\propto 2^{28} \approx 2.68\times10^{8}$步，而经典算法则约需 $2^{55} \approx 3.6\times10^{16}$步。若每秒计算十亿次，经典计算需 11 年，而 *Grover* 算法只需 3 秒钟。详细可见有关文献。另外，可以证明，这里 *Grover* 算法是搜寻算法中最快的算法，它比任何可能的量子搜寻算法都要好【6】。

Grover 算法是最能体现量子并行性的算法，这种算法以它在遍历搜寻问题上的应用而著名，但这并不表示它不能用来处理其它问题。事实上，*Grover* 算法在解决经典算法难题方面的应用要比 *shor* 算法较为广泛。从原则上讲，它可以解决诸如“求解困难而验证容易”的经典慢算法问题。但量子计算的实验实现方面仍然存在一些重要的问题【7】。

练习题

[15.1] 已知文中 $y=11, N=21$的 *Shor* 算法例子里 $r=6$，现再直接算出 *Euler* 的 *phi*-函数 $\varphi(N)$值，验证文中所说 *Euler* 定理的充分性。

[15.2] 检验 $5^r = 1 \ mod\ 21$的周期 r是 6。

[15.3] 现拟一个应用孙子定理的类似问题：“今有物不知其数，三三数之剩 2，五五数之剩 4，七七数之剩 1。问物几何？”利

用程大位歌诀给出计算结果。

[15.4] 验证：$\sum_{s=0}^{q-1}\exp\left(-2\pi i\, sk/q\right)=q\delta_{k0}$。

[15.5] 变换 $\hat{T}|n\rangle=\frac{1}{\sqrt{2}}\left(|0\rangle+(-1)^n|1\rangle\right),\ (n=0,1)$是 *Walsh-Hadamard* 变换（32）式的特例。针对双 *qubit* 回路计算 $\left(\hat{T}_1\hat{T}_2\right)\hat{U}_{CNOT}\left(\hat{T}_1\hat{T}_2\right)|j,k\rangle_{12}$。

参考文献

【1】 *A.Ekert,et.al., Rev. of Mod. Phys.,* 68（1996）733。

【2】 *M.A.Nielsen and I.L.Chuang, Quantum Computation and Quantum Information, Cambridge University Press,* 2000, p.216, p.637。

【3】 *P.W.Shor, Proc. of the 35-th Annual Symposium on the Foundations of Computer Science, edited by S. Goldwasser(IEEE Computer Science, quantum ph*/9508027)。

【4】*L.K.Grover, PRL,* 79(1997)325; *L.K.Grover, Science,* 280(1998) 228;也见 *M.A.Nielsen and I.L.Chuang* 《*Quantum Computation and Quantum Information*》，p.248。

【5】 参见比如，陈景润，《初等数论》，科学出版社，1978．P. 83。

【6】*J. Preskill, Lecture Notes for Physics* 229: *Quantum Information and Computation, CIT,* 1998.9。

【7】 王力军，《量子计算中有待解决的几个基本问题》，全国

高校量子力学研究会年会上的报告，南京大学，2001，4。