Quantum Computation

Yongjian Han

University of Science and Technology of China

Spring 2011

中国科学技术大学

Overview

In the following, we will introduce quantum algorithms (Such as, Shor factoring algorithm, Grover search algorithms),three quantum computing models: quantum Circuit model,quantum adiabatical computation model and one-way computer. We also introduce how to realize real quantum computation in ion trap and linear optics. At last, we will introduce quantum error-correction theory and fault-tolerant quantum computation.

Lecturer

YongJian Han (smhan@ustc.edu.cn)
Office: Key Lab of Quantum Information, Room 501

Contents

- Introduction of quantum computation and computation complexity
- quantum algorithm (I): Duetch-Jozsa algorithm and Grover algorithm
- quantum algorithm (II): Simon algorithm,quantum Fourier Transform and Shor factoring algorithm
- Three quantum computation models (I): computation circuit
- Three quantum computation models (II): quantum adiabatical computation
- Three quantum computation models (III): one-way computer

Continue contents

- Toward real quantum computation: physical system (I) linear optics
- Toward real quantum computation: physical system (II) ion trap
- Toward real quantum computation: noise and decoherence
- Toward real quantum computation: Error-correction (I) stabilizer code
- Toward real quantum computation: Error-correction (II) topological quantum computation
- Toward real quantum computation: fault-tolerant computation

Website

http://lqcc.ustc.edu.cn/simulate/upload

quantum computing history

- Church-Turning machine is universal for any algorithm process
- any algorithm process can be efficiently simulated by probabilistic Turning machine
- D. Deutch proposed a quantum Turning machine on the same way
- R. Feynman suggested to simulate quantum system by a simple quantum system
- several algorithms show the power of quantum computation: Deutch-Jozsa algorithm, Simon algorithm, quantum QFT, Shor algorithm, Grover algorithm
- Can quantum computation beyond classical computation?

computation complexity

The classical computation complexity classes are based on the computing model of classical Turning model. Since the universality of this computation model, the classes are well defined.

- How to define the difficulty to solve a problem? the criterion of the complexity of a problem is based on time scaling
  - P problem: a Problem can be solved in polynomial time
  - NP problem: a problem can be checked in polynomial time
- the most important problem is the relation between P and NP.
- The most difficulty problems in NP are NPC problems, NPC problems are also universal. The first proven NPC problem is 3-SAT problem
  - 3-SAT problem is a Boolean decision problem: Do exist logic values make all the clauses true?

computation complexity

Since the quantum computer can factor a integer more efficiently than the classical computer, Can we solve all the problems in NP class efficiently? We have some hints in this problem

- Shor algorithm is more efficient than any classical algorithm and can factor a integer efficiently.
  - Factering problem is a NP problem
  - Open question: whether Factering problem is a NPC problem
- Quantum computation is based on parallelism of quantum mechanics while the finite parallelism can not reduce the source scaling based on a search-based methodology
- But quantum computation have a different structure (we will see in the 3 computation models), this failed method do not exclude the quantum computer can solve NP problem efficiently.

computation complexity

Beside the P and NP classes ,There are also some other classes defined on classical computation model

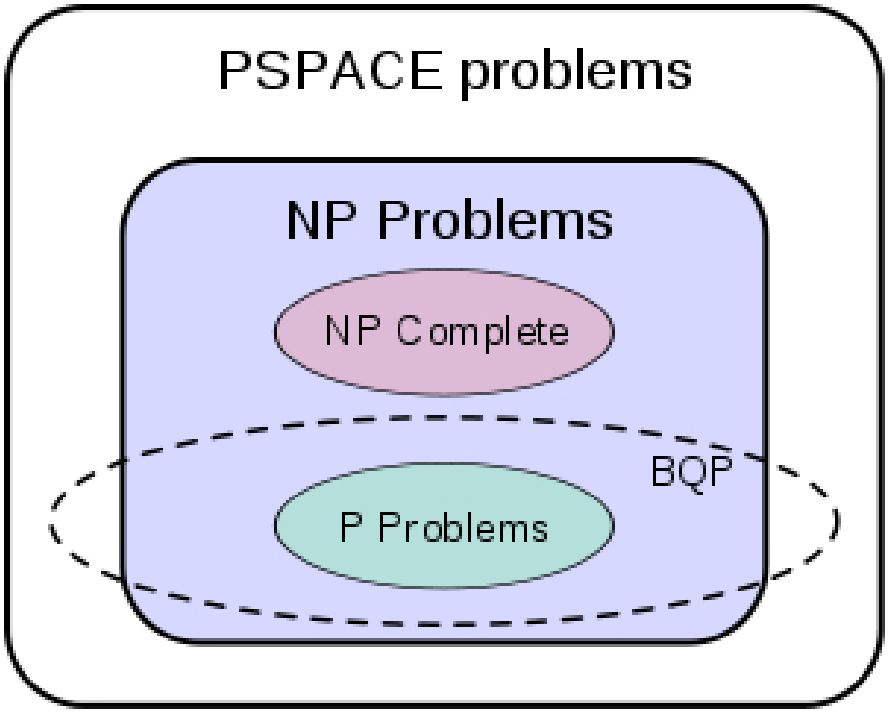
- PSPACE class: the problems can be solved by polynomial space source
- BPP class: the problems can be solved using randomized algorithms in polynomial time if a bounded probability of error is allowed

Similar as the classical computation complexity, we can define quantum complexity.

- BQP class: the problems can be solved by quantum computer in polynomial time if a bounded probability of error is allowed

computation complexity

We have defined several computation classes, the key problem in computational complexity theory is to find the relation between them. The widely believed relation as following figure though most of them unproved.



Deutsch-Josza algorithm

The quantum algorithm is designed to using the quantum parallelism to accelerate classical computation. In general, the algorithm is divided in the following steps:

- preparation a initial state, at most time it is $|+\rangle|+\rangle|+\rangle...|+\rangle$
- a set of operations to evolution the initial state
- read out the result

Generally, the output of the result is not unique, the right answer appear as some finite probability

Deutch-Josza problem

distinguish a function $f : \{0,1\}^n \to \{0,1\}$, a balance one or a constant one?

Deutsch-Josza algorithm

The quantum computer can solved this problem once.

$$|0\rangle^n|1\rangle \rightarrow (1/2^{n/2}.\sum_{x=0,1,..,2^n-1}|x\rangle).1/\sqrt{2}(|0\rangle - |1\rangle) \quad (1)$$

- operating Hadamard transformation on every qubit
- the last qubit served as a register qubit to save the information

$$\rightarrow (1/2^{n/2}.\sum_{x=0,1,..,2^n-1}(-1)^{f(x)}|x\rangle).1/\sqrt{2}(|0\rangle - |1\rangle) \quad (2)$$

- operating a black box- n-controlled unitary U_f transformation on- this state

$$U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle \quad (3)$$

Deutsch-Josza algorithm

$$\rightarrow (1/2^n . \sum_{x=0,1,..,2^n-1} \sum_{y=0,1,..,2^n-1} (-1)^{f(x)} (-1)^{x.y} |y\rangle).1/\sqrt{2}(|0\rangle - |1\rangle) \tag{4}$$

- operating Hadamard transformation on the qubit except the last one (register one)

$$H^n : |x\rangle \rightarrow \prod_{i=1...n} (1/\sqrt{2} . \sum_{y_i=0,1} (-1)^{x_i \cdot y_i} |y_i\rangle) \tag{5}$$

$$= 1/2^{n/2} . \sum_{y=0,1,..,2^n-1} (-1)^{x.y} |y\rangle \tag{6}$$

- $x.y$ is the bitwise AND

Deutsch-Josza algorithm

Now we focus on the coefficient of $|y\rangle$, if f is constant

$$(-1)^{f(x)}(1/2^n \sum_{x=0,1,..,2^n-1} (-1)^{x.y}) = (-1)^{f(x)}\delta_{y,0} \quad (7)$$

When measure the n-qubit system, it will be definitely on state $(|0\rangle)^n$ however, if f is balance,we measure the n-qubit system, the probability to find the state $(|0\rangle)^n$ is

$$1/2^n. \sum_{x=0,1,..,2^n-1} (-1)^{f(x)} = 0 \quad (8)$$

Deutsch-Josza algorithm

The computation complexity of this problem can be fefined in two different ways

- If to judge this problem exactly, it need to check $2^{n-1}+1$ numbers
- on the other hand, if just to require the probability of error under some bound ε, the scaling of this problem is

$$k \sim 1/2.\lg(1/\varepsilon) \tag{9}$$

Grover search algorithm

The former question is a bit artificial. Now we turn to a widely used problem: search a certain item (such as ω) to fit some condition in a database without order.

- Oracle

To submit a query x to the oracle and it tells us whether $x = \omega$ or not. It return as

$$f_\omega(x) = 0, x \neq \omega \tag{10}$$

$$f_\omega(x) = 1, x = \omega \tag{11}$$

Grover search algorithm

At first we define a black box similar in Deutch Josza algorithm

$$U_{f_\omega} : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle \tag{12}$$

This black box operating as

$$U_{f_\omega} : |x\rangle(|0\rangle - |1\rangle) \rightarrow (-1)^{f_\omega}|x\rangle(|0\rangle - |1\rangle) \tag{13}$$

If ignore the second register, the transformation will be

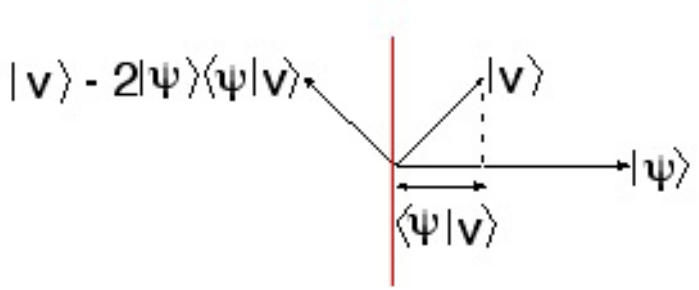
$$U_\omega : |x\rangle \rightarrow (-1)^{f_\omega}|x\rangle \tag{14}$$

or

$$U_\omega : I - 2|\omega\rangle\langle\omega| \tag{15}$$

Grover search algorithm

Geometry interpretation of the Oracle



- We need to define another transformation

$$U_s : 2|s\rangle\langle s| - I \tag{16}$$

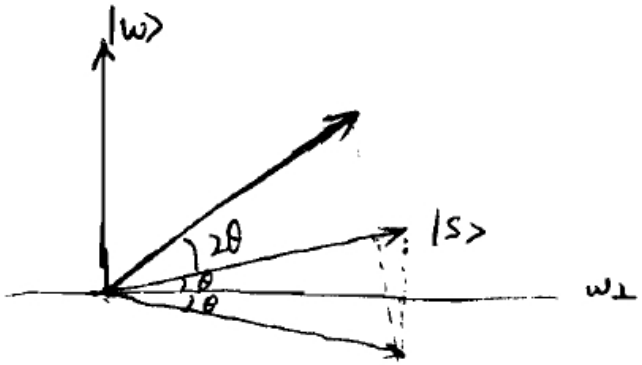
where $|s\rangle = 1/\sqrt{N}.\sum_{x=0,1,..2^N-1} |x\rangle$

- now to define the operator of grover algorithm

$$R_{grov} = U_s.U_\omega \tag{17}$$

Grover search algorithm

Geometry interpretation of Grover operator



- The R_{grov} operator rotate the state $|s\rangle$ to the final state $|\omega\rangle$ as 2θ in the plane determined by $|s\rangle$ and $|\omega\rangle$, here $sin\theta = 1/\sqrt{N} = \langle s|\omega\rangle$
- Since it will rotate 2θ by each step, T step will receive $(1+2T)\theta$. It should be optimized to close $\pi/2$

$$(2T+1)\theta \approx \pi/2 \Rightarrow 2T+1 \approx \pi/2\theta \tag{18}$$

for large N,it can be simplified as $T = \pi/4.\sqrt{N}$

Grover search algorithm

- the scaling of the search problem is $\sqrt{N}$, instead of N which is the scaling of the classical algorithm
- the probability of get the right answer is about $\sin^2((2T+1)\theta)) = 1 - O(1/N)$
- search problem is a widely used problem, many problem can be translate to this problem. So the speedup has some kind of universal character for NP problem.
- Grover algorithm is the optimal search algorithm for unsorted database
- For multiple solution, the algorithm can also work. But the number of the solution should be known before, and the state $|\omega\rangle$ should be changed as $|\tilde{\omega}\rangle = 1/\sqrt{r}.(\sum_{i=0,1,..r} |\omega_i\rangle)$ where r is the solution number. (Exercise)

Summary

- Inputs: (1) a black box: $U|x\rangle|q\rangle = |x\rangle|q \oplus f(x)\rangle$,where $f(x) = 0$ when $0 \leq x \leq 2^n$ except $x = \omega$; (2)$n + 1$ qubits in the state $|0\rangle$
- Output: ω
- Runtime: $O(\sqrt{2^n})$ operations and Succeeds with probability $O(1)$.

Summary continue

- Procedure

$|0\rangle^{\oplus n}|0\rangle$ initial state

$$\rightarrow |\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0,1,..,2^n-1} |x\rangle[\frac{|0\rangle - |1\rangle}{\sqrt{2}}]$$

apply H to the first n and HX to the last qubits

$$\rightarrow [R_G]^{\otimes R}|\psi\rangle \approx |\omega\rangle[\frac{|0\rangle - |1\rangle}{\sqrt{2}}]$$

apply Grover iteration R times

$\rightarrow \omega$ measure the first n qubits

Simon algorithm

Problem: Known the function $f(x)$ has a period, that is

$$f(x) = f(y) \iff y = x \bigoplus a \tag{19}$$

where $\bigoplus$ is a bitwise XOR operation.

- For classical case, If we calculate $2^{n/4}$ number of the function, the pair can be constructed is $2^{n/2} - 2^{n/4}$. By these data, the probability to find the period a is less than

$$2^{-n} * (2^{n/2} - 2^{n/4}) = 2^{-n/2} - 2^{-3n/4} < 2^{-n/2} \tag{20}$$

  which is exponentially small
- For quantum case, we also define a unitary black box

$$U_f : (\sum_{x=0,1,..,2^n-1} |x\rangle)|0\rangle \to (\sum_{x=0,1,..,2^n-1} |x\rangle)f(x)\rangle \tag{21}$$

Simon algorithm

- measure the second register,assume the result is $f(x_0)$. Since the period of a, the rest register will be a superposition state

$$1/\sqrt{2}(|x_0\rangle + |x_0 + a\rangle) \tag{22}$$

- operate Hadamard transformation on every qubit

$$H^n : 1/\sqrt{2}(|x_0\rangle + |x_0 + a\rangle) \tag{23}$$

$$\rightarrow 1/2^{(n+1)/2} \sum_{y=0,1,\ldots,2^n-1} ((-1)^{x_0 y} + (-1)^{(x_0 \oplus a).y})|y\rangle \tag{24}$$

$$= 1/2^{(n-1)/2} \sum_{a.y=0} ((-1)^{x_0 y}|y\rangle \tag{25}$$

- measure on all the register will find some y which satisfy $a.y = 0$. Repeat all the processes to get enough independent $y : \{y_1, y_2, .., y_n\}$. Then we can find the period a by solving the equations.The scaling of the number of repeating is just n.

QFT algorithm

Problem: realize a quantum Fourier transformation, which is

$$\sum_x f(x)|x\rangle \to \sum_y (1/\sqrt{N} \sum_x e^{2\pi i x y/N} f(x))|y\rangle \tag{26}$$

- The naive way to realize this transformation is write the transformation as a $N \times N$ matrix, this matrix product a vector. So the scaling is $O(N^2)$
- the well-known classical FFT algorithm can be done as scaling $O(N \log N)$. Suppose $N = 2^n$,x and y can be write in binary form as

$$x = (x_{n-1}x_{n-2}...x_1x_0) \tag{27}$$

$$y = (y_{n-1}y_{n-2}...y_1y_0) \tag{28}$$

QFT algorithm

- Since $e^{2\pi ik}$ is a period function. A integer adding to k has no contribution.

$$xy/2^n \equiv y_{n-1}(.x_0)+y_{n-2}(.x_1x_0)+...+y_0(.x_{n-1}x_{n-2}...x_0) \quad (29)$$

where

$$.x_2x_1x_0 = x_2/2 + x_1/2^2 + x_0/2^3 \quad (30)$$

- each coefficient (total N)

$$\tilde{f}(x) = 1/\sqrt{N}\sum_y e^{2\pi ixy/N} f(y) \quad (31)$$

can be calculated by $y_k = 0, 1$ in time of scaling n.

QFT algorithm

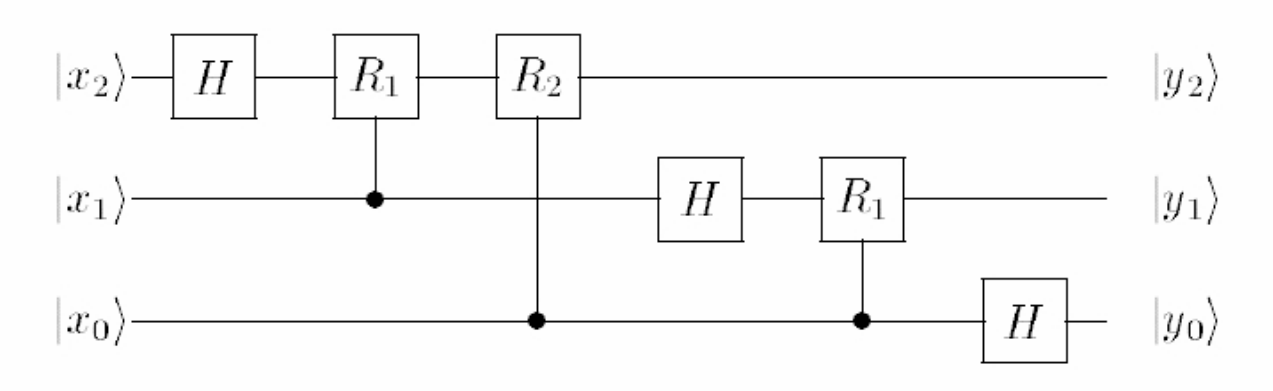
- In quantum case, It can be more efficient by quantum parallelism

$$QFT : |x\rangle \rightarrow 1/\sqrt{N} \sum_{y} e^{2\pi i x y/N} |y\rangle \tag{32}$$

$$= 1/\sqrt{2^n}(|0\rangle + e^{2\pi i(.x_0)}|1\rangle)(|0\rangle + e^{2\pi i(.x_1 x_0)}|1\rangle) \tag{33}$$

$$...(|0\rangle + e^{2\pi i(.x_{n-1}x_{n-2}...x_0)}|1\rangle) \tag{34}$$

- this operation can be implemented by the following way



QFT algorithm

- where H is a Hadamard gate and R_d acts as

$$H : |x_k\rangle \rightarrow 1/\sqrt{2}(|0\rangle + e^{2\pi i(.x_k)}) \tag{35}$$

$$R_d = \{1, 0; 0, e^{i\pi/2^d}\} \tag{36}$$

  where d is the distance between the qubits.
- the source for this circuit need n Hadamard gate and $n(n-1)/2$ Control-R gate, so the scaling is $O((logN)^2)$

finding the period

Problem: A function

$$f : \{0,1\}^n \to \{0,1\}^m \tag{37}$$

there is a unknown positive period $r : 1 \ll r \ll 2^n$, that is,

$$f(x) = f(x + mr) \tag{38}$$

where m is a integer make x and $x + mr$ lie in $\{0, 1, 2, ..., 2^n - 1\}$. Our task is to find the period r.

- This problem is very hard in classical case
- however, Using the former QFT algorithm, there is a polynomial time algorithm to solve this problem.
- to compare with the Simon's problem

finding the period

- similar as Simon algorithm, define a quantum black box

$$U_f : (\sum_{x=0,1,..,2^n-1} |x\rangle)|0\rangle \rightarrow (\sum_{x=0,1,..,2^n-1} |x\rangle)f(x)\rangle \tag{39}$$

- measure the output register and get result $f(x_0)$ (where $0 \leq x_0 \leq r$. The input register will in the superposition state

$$1/\sqrt{A} \sum_{j=0,1,2..,A-1} |x_0 + jr\rangle \tag{40}$$

where $N - r \leq x_0 + (A-1)r < N$, that is,

$$A - 1 < N/r < A + 1 \tag{41}$$

- similar as the Simon algorithm we do not measurement the state directly, we should do QFT on the former superposition state

finding the period

we can get the following expression by QFT

$$1/\sqrt{NA} \sum_{y=0,1,..,N-1} e^{2\pi i x_0 y} \sum_{j=0,1,..,A-1} e^{2\pi i j r y/N} |y\rangle \tag{42}$$

- Then we measure in the $|y\rangle$ basis, the probability to get the outcome y is

$$Prob(y) = |1/\sqrt{NA} e^{2\pi i x_0 y} \sum_{j=0,1,..,A-1} e^{2\pi i j r y/N}|^2 \tag{43}$$

$$= A/N|(\sum_{j=0,1,..,A-1} e^{2\pi i j r y/N})/A|^2 \tag{44}$$

- In the following, we will give a estimate of the probability, which is

$$Prob(y) \geq (4/\pi^2).1/r \tag{45}$$

finding the period

Estimate $Prob(y)$

- for any geometry series

$$\sum_{j=0,1,...A-1} e^{i\theta j} = (e^{iA\theta} - 1)/(e^{i\theta} - 1) \tag{46}$$

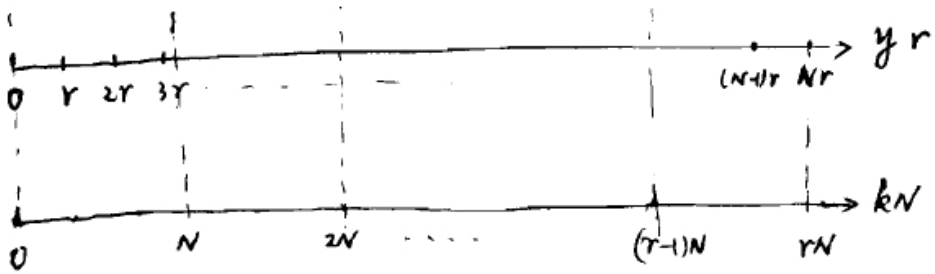
  where $\theta = 2\pi.yr(modN)/N$

- suppose the parameter y satisfies the condition

$$-r/2 \leq yr(modN) \leq r/2 \tag{47}$$

$$\text{or } -\pi r/N \leq \theta_y \leq \pi r/N \tag{48}$$

  there are r such y in $\{0, 1, 2, ..., N-1\}$



finding the period

- under the former condition, the probability can be find

$$Prob(y) \geq (4/\pi^2).1/r \tag{49}$$

  - By some simple triangle function calculation,$|1 - e^{i\theta}| = 2|\sin\theta/2| \leq |\theta|$
  - by the character of sin function $|1 - e^{iA\theta}| \geq 2A|\theta|/\pi$
  - Using the former inequalities, it can be derived that

$$|(e^{iA\theta} - 1)/(e^{i\theta} - 1)| = |(e^{i(A-1)\theta} - 1)/(e^{i\theta} - 1) + e^{i(A-1)\theta}| \tag{50}$$

$$\geq |(e^{iA\theta} - 1)/(e^{i\theta} - 1)| - 1 \tag{51}$$

$$\geq 2(A-1)/\pi - (1 + 2/\pi) \tag{52}$$

  - from the equation(41) and (48), we can only get $(A-1)\theta \leq \pi$, so consider the convex of the distance function.

finding the period

- since there are r values of y in Eq.(40), we have probability $4/(\pi^2)$ to get some y_0 satisfy (the equation (47))

$$|y_0 r - kN| \leq r/2 \tag{53}$$
$$\Rightarrow kN/r - 1/2 \leq y_0 \leq kN/r + 1/2 \tag{54}$$
$$\Rightarrow k/r - 1/2N \leq y_0/N \leq k/r + 1/2N \tag{55}$$

- As so far, the measurement gets y_0/N, But the parameters k and r is still unknown.
- if $r < \sqrt{N}$, k/r can be uniquely determined by y_0/N. This can be determined by the continued fraction method.

finding the period

- continued fraction expansion: Suppose s/r is a rational number such that

$$|s/r - \varphi| \leq 1/2r^2 \tag{56}$$

then s/r is a convergent of the continue fraction for φ and thus can be computed in $O(\log N^3)$ operations using continued fraction algorithm.

- continued fraction algorithm: if

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ldots}}} \tag{57}$$

(For any rational number, it can be expressed as the continued fraction form as $[a_0, a_1, a_3, ...]$)

finding the period

- nth convergent of x is $[a0, a_1, a_2, .., a_n]$, Using this expression, we can get the nth rational number convergence p_n/q_n as

$$p_0 = a_0 \quad p_1 = a_1 a_0 + 1 \quad ... \quad p_n = a_n p_{n-1} + p_{n-2} \tag{58}$$

$$q_0 = 1 \quad q_1 = a_1 \quad ... \quad q_n = a_n q_{n-1} + q_{n-2} \tag{59}$$

- the condition $|k/r - y_0/N| \leq 1/2N \leq 1/2r^2$ satisfies the requirement of continued fraction expansion, so we can can find the unique values k and r with no common factor
- Since k take from $\{1, 2, .., r-1\}$, the probability of k and r without common factor is $\phi(r)/\gamma > \frac{e^{-\gamma}}{\log\log N}$,where $\phi(r)$ is the Euler function and γ is a Euler constant about 0.577

finding the period

- combine the former probabilities to find the probability to find the right r is about $\frac{4}{\pi^2}\frac{e^{-\gamma}}{\log\log N}$
- the scaling of this problem is just $\log\log N$
- any function which can be calculated in polynomial time, there is a algorithm to find its period in polynomial time

- Summary
  - Input: (1)a black box: $U|x\rangle|y\rangle = |x\rangle|y \bigoplus f(x)\rangle$;(2)$t = t + 1$ qubit initialized to $|0\rangle$ where $t = O(L + \log(1/\epsilon))$ and $r < 2^L$
  - Output: the least r such that $f(x) = f(x + r)$
  - Run time: one use U and $O(L^2)$ operations, success probability $O(1)$

- Summary continue
  - procedure:

$$|0\rangle|0\rangle \qquad \text{initial state} \tag{60}$$

$$\rightarrow \frac{1}{\sqrt{2^t}} \sum_{x=0,1,..,2^t-1} |x|\rangle|0\rangle \qquad \text{create superposition} \tag{61}$$

$$\rightarrow \frac{1}{\sqrt{2^t}} \sum_{x=0,1,..,2^t-1} |x|\rangle|f(x)\rangle \qquad \text{apply U} \tag{62}$$

$$\simeq \frac{1}{r\sqrt{2^t}} \sum_{l=0,1,2..,,r-1} \sum_{x=0,1,..,2^t-1} e^{2\pi i l x/r}|x|\rangle|\tilde{f}(l)\rangle \tag{63}$$

$$\rightarrow \frac{1}{\sqrt{r}} \sum_{x=0,1,..,r-1} |l\tilde{/}r|\rangle|\tilde{f}(l)\rangle \qquad \text{apply QFT} \tag{64}$$

$$\rightarrow l\tilde{/}r \qquad \text{measure the first register} \tag{65}$$

$$\rightarrow r \qquad \text{apply continue fraction algorithm} \tag{66}$$

factoring

- Problem: Given a composite n-bit number $N = pq$, to find its prime factors.
- this problem can be reduced to find the period of a function
  - select a number a randomly, and find $GCD(a, N)$ (this can be done efficiently by standard Euclidean algorithm)
  - suppose $GCD(a, N) = 1$, or $GCD(a, N)$ will be a nontrivial factor of N
  - a^r will generate a finite cycle group and there is a r satisfied $a^r \equiv 1(\mod N)$. So, the function $f_{N,a}(x) = a^x(\mod N)$ has a period.
  - Function $f_{a,N}$ can be efficiently calculated
- we use the quantum algorithm to find the period r of $f_{a,N}(x)$.

factoring

- if r is even, then N can divide $(a^{r/2}+1)(a^{r/2}-1)$.
- N does not divide $a^{r/2}-1$, or $a^{r/2} \equiv 1(\mod N)$, that is, $r/2$ is a period.
- if N does not divide $a^{r/2}+1$, then $GCD(N, a^{r/2} \pm 1)$ are nontrivial factor of N. It is done.

There are two condition to make the algorithm success.

- r is even
- N does not divide $a^{r/2}+1$

there have $1/2$ probability to satisfy these conditions.

factoring

- Summary
  - Input: A composite number $N = pq$, where p, q are co-prime
  - Output: a nontrivial factor of N
  - Run time: $O(\log(N)^3)$ operations, success probability $O(1)$
  - procedure:
    - randomly choose a in the range 1 to $N-1$, If $GCD(a, N) > 1$, then return the factor $GCD(a, N)$.
    - using quantum period algorithm to get the period r , which make $a^r \equiv 1(\mod N)$
    - if r is even and $a^(r/2) \equiv -1(\mod N)$, then compute $GCD(a^{r/2} - 1, N)$ and $GCD(a^{r/2} + 1, N)$, and test to see if one of them is nontrivial. If so return it. otherwise, the algorithm fails.

other application of QFT

- quantum order-finding algorithm
- quantum phase estimation
- discrete logarithm
- hidden subgroup problem

introduction

- The favorite method to understand the world is to divide it into some parts

a physical system	a software
cell	a code for a single task
atom	a code for a single function
basic particles	basic gates

- any algorithm can be constructed by these basic gates
- A algorithm is a circuit of basic gates
- This is the standard model of classical computation

basic classical gates

The classical logical gates

- NOT, AND, OR and $COPY$ are basic logical gates of the classic computation. Any function $f(x) : \{0,1\}^n \to \{0,1\}$ can be calculated by these connectives.
  - $OR(\bigvee) : x \bigvee y = x + y - x.y$
  - $AND(\bigwedge) : x \bigwedge y = x.y$
  - $NOT(\bar{}) : \bar{x} = 1 - x$
  - $COPY : x \to xx$
- $NAND$ or NOR and $COPY$ are another set of basic gates
  - $NAND : x \uparrow y = 1 - x.y$
  - $NOT : \bar{x} = x \uparrow x$
  - $AND : x \bigwedge y = (x \uparrow y) \uparrow (x \uparrow y)$
  - $OR : x \bigvee y = (x \uparrow x) \uparrow (y \uparrow y)$

basic quantum gates

- If a constant bit can be prepared, the elementary gate can be reduced to one,such as NAND/NOT: $(x, y) \rightarrow (1 - x, 1 - x.y)$, Since the preparation of initial state play the similar role as COPY gate.
  - ignore the first qubit it will be a NAND gate
  - set $y = 1$, it will be a COPY gate

the classical algorithm, code, and software will constructed by these elementary gates. The complexity of the problem is dependent on the complexity of the circuit.

- COPY play an important role in classical computation, while it can not work in quantum computation.
- One of the difference between classical and quantum gate is that quantum gate is reversible. Generally, a irreversible computation can be realized by reversible computation and have the same computation complexity [C. Bennett claimed].

reversible computation gates

to find the universal gates for reversible computation.

- The reversible gates $f : \{0,1\}^n \to \{0,1\}^n$, can be regard as a permutation of the 2^n strings
- there are only 2 gates are reversible for 1-bit gate, I and NOT.
- there are $(2^2)!$ gates are reversible for two-bit gate,such as,$XOR : (x, y) \to (x, x)$.
- one and two-bit reversible gate is not universal for reversible computation. The are linear. They can not calculate some nonlinear gate, such as Toffoli gate.
- three-bit Toffoli gate is nonlinear and define as

$$\theta^{(3)} : (x, y, z) \to (x, y, z \bigoplus x.y) \tag{67}$$

- Toffoli gate is universal for reversible computation.[constructive proof]

basic quantum gates

A quantum computation with quantum Circuit model is divided in the following sections

- a finite number n of qubits are initially set to the value $|00..0\rangle$
- a quantum circuit constructed from a finite number of quantum gates work on these qubits.
- a Von Neumann measurement of some qubits is performed, projecting each onto the basis $\{|0\rangle, |1\rangle\}$ The outcome of this measurement is the result of the computation.

Several comments on this computation model

- the Hilbert space prefer to be decomposited into local subsystems, the basic gate supposed to only operate on their neighbors

basic quantum gates

- though the Unitary transformation form a continuum, To pursue the universality, we suppose to find some discrete universal set
- The general measurement is POVM, But we can translate this measurement to Von Neumann measurement on some extended system.
- theoretically, measurement can be done on any basis, But all this basis can be translated to the computation basis by unitary transform which can be included in the circuit.
- In general, we can measurement during the computation, but all these measurement can be postponed to the end.

Now we turn to find some universal basic gates.

basic quantum gates

Before consideration the universal gate, there are several basic facts about quantum gates

- a 'generic' k-qubit U with eigenvalues $\{e^{i\theta_1}, e^{i\theta_2}, .., e^{i\theta_{2^k}}\}$
  - θ_i are irration multiple of π.
  - all the θ_is are incommensurate, that is, each θ_i/θ_j is also irrational.

  U^n define a point in a 2^k-dimensional torus. Under the former condition, as n ranges over positive integer values, these points densely fill the whole torus.
- Construct a new gate U' by a Swapping gate P and an old gate U by

$$U' = PUP^{-1} \tag{68}$$

basic quantum gates

- completing the Lie algebra: known,two generic unitary transformation e^{iA} and e^{iB}, to come arbitrary close to unitary transformation $e^{i(\alpha A+\beta B}$ and $e^{i[A,B]}$

$$\lim_{n\to\infty}(e^{i\alpha A/n}e^{i\beta B/n})^n \tag{69}$$

$$=\lim_{n\to\infty}(1+i/n(\alpha A+\beta B))^n \tag{70}$$

$$=e^{i(\alpha A+\beta B} \tag{71}$$

So $e^{i(\alpha A+\beta B}$ is reachable.

basic quantum gates

- we consider the Lie algebra case now

$$\lim_{n\to\infty}(e^{iA/\sqrt{n}}e^{iB/\sqrt{n}}e^{-iA/\sqrt{n}}e^{-iB/\sqrt{n}})^n \tag{72}$$

$$= \lim_{n\to\infty}[(1+iA/\sqrt{n}-A^2/2n)(1+iB/\sqrt{n}-B^2/2n)$$

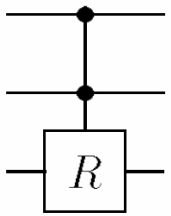
$$(1-iA/\sqrt{n}-A^2/2n)(1-iB/\sqrt{n}-B^2/2n)]^n \tag{73}$$

$$= \lim_{n\to\infty}[1-\frac{AB-BA}{n}]^n \tag{74}$$

So we can complete a Lie algebra by this method

basic quantum gates

Deutsch's three-bit universal quantum gate



The matrix R applies to the third qubit when the first two qubits are 1,otherwise it acts trivially.

$$R = -iR_x(\theta) = (-i)exp(i\theta\sigma_x/2) \tag{75}$$

$$= (-i)(cos\frac{\theta}{2} + i\sigma_x sin\frac{\theta}{2}) \tag{76}$$

it is a rotation by θ around x-axis, where θ is a angle incommensurate with π

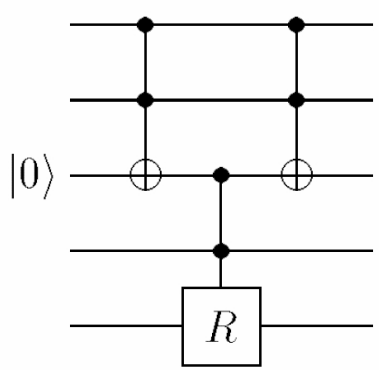
basic quantum gates

we investigate the universality of Deutsch gate

- Due to the character of θ the $(4n+1)st$ power can come as close as we want to σ_x
- the toffoli gate can be reachable by Deutch gate, so Deutsch gate is universal for classical computation. Also can implement reversible function by adding some auxiliary qubits.
- In the standard basis of three qubits, the Deutsch gate generator is: $(\sigma_x)_{67} = I \bigotimes I \bigotimes \sigma_x$
- Toffoli gate (can generate by Deutsch gate) can generate any permutation operation P of any two basis, then any generator like $(\sigma_x)_{mn}$ can be reached by $P\sigma_x)_{67}P^{-1}$
- Furthermore, we can reach $(\sigma_y)_{mn}$ can be reached by $i[(\sigma_x)_{mk}, (\sigma_x)_{kn}]$

basic quantum gates

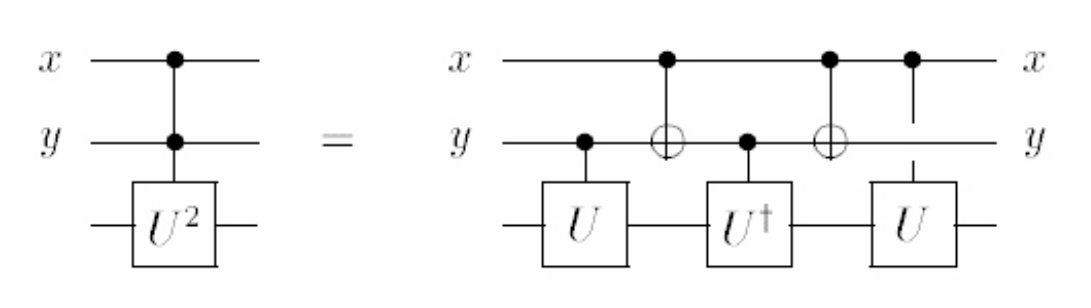
- then we can reach the generator $(\sigma_z)_{mn}$ can be reached by $i[(\sigma_x)_{mk}, (\sigma_y)_{kn}]$
- Now the whole generator of $SU(8)$ can be generated.
- at last we can generate a $n-qubit$ Toffoli gate by the three-qubit Toffoli gates. For example: $4-qubit$ Toffoli gate can be generated by



basic quantum gates

In fact, in universal quantum computation, Three-qubit gates are not necessary. There are universal two-qubit gates.

- We need only prove that there is a two-qubit gates can generate a 3-qubit Deutsch gate.
- we can construct a controlled-U^2 gate from controlled-U gates by



  where the power of the U determined by $y-(x \bigoplus y)+x=2xy$.
- So we can construct Deutsch gate from from the controlled-U controlled U^{-1} and controlled-NOT gates, where

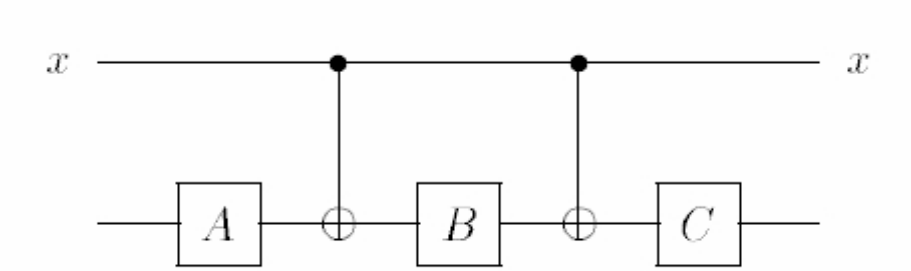
basic quantum gates

$$U = e^{-i\pi/4}.R_x(\frac{\theta}{2}) \tag{77}$$

- when the parameter θ/π is irrational, U^{-1} and σ_x gates came as close as we want. Then Deutsch gate can be generated and it is universal

In fact, almost all the two-qubit gates (generic gates) are universal. The prove is similar as the universal of Deutsch gates.

- C-NOT and one-qubit gates form a universal set
- any controlled-U can be constructed by C-NOT and one-qubit gates by



basic quantum gates

where

$$ABC = 1 \qquad A\sigma_x B\sigma_x C = U \tag{78}$$

The rest problem for quantum circuit is to design optimal circuit to certain task under a universal set. This problem is very difficult in general, a generic algorithm is proposed to this problem.

introduction

- the former circuit model is a based on the classical Von Neumman computation structure, there are many classical methods can be introduce into quantum situation.
- quantum computation based on different physical system, it has its own character
- quantum adiabatical computation is a model which is close related to many-body physics, it convert the computation problem to the gap of a many-body physics
- quantum adiabatical computation is a good model to consider the complexity problem.
- one-way computer is another model to consider the character of quantum system. It is very useful to consider fault-tolerant computation.

adiabatic theorem

the evolution of a system is according to Schordinger Equation

$$ih\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle \tag{79}$$

the adiabatic theorem tells us how to follow this evolution in the case that $H(t)$ is slowly varying

- at every time, the Hamiltonian $H(t)$ has its eigenstates and eigenvalues

$$H(s)|l;s\rangle = E_{l(s)}|l;s\rangle \tag{80}$$

  where l is the quantum number of time s
- suppose the system in state $|l=0;s=0\rangle$ at the time $t=0$
- Adiabatic theorem: if the gap between the two lowest levels, $E_1(s)-E_0(s)$, is strictly greater than zero for all $0 \le s \le 1$, the state will stay on $|0;s\rangle$ when the $H(t)$ varying enough slow

adiabatic theorem

- define:

$$g_{min} = \min_{0 \le s \le 1} (E_1(s) - E_0(s)) \tag{81}$$

where $\varepsilon = \max_{0 \le s \le 1} |\langle l = 1; s| \frac{dH}{ds} |l = 0; s\rangle|$, the evolution time can be estimate by

$$T \ge \frac{\varepsilon}{g_{min}^2} \tag{82}$$

- Generally,ε is a stable number, and the time T is determined by g_{min}.

satisfiability problem

- An n-bit instance of satisfiability is a formula $C_1 \bigwedge C_2 \bigwedge ... \bigwedge C_n$ where each clause C_a is True or False depending on the values of some subset of the bits. For a single clause, involving only a few bits. Our task is to judge whether there is an assignment that satisfies all n clauses.
- using 3-SAT problem as an example. 3-SAT problem is a proved NPC problem.
- Every Clause C is associated with the 3 bits, and we can define an energy function

$$h_C(z_{iC}, z_{jC}, z_{kC}) = 0 \tag{83}$$

if (z_{iC}, z_{jC}, z_{kC}) satisfies clause C , else it will be 1

satisfiability problem

- the total energy of the problem will be

$$h = \sum_C h_C \tag{84}$$

clearly,$h \geq 0$and if and only if $(z_1, z_2, ..., z_n)$ satisfies all the clauses.

To translate this problem to quantum problem, we label a $1/2$ spin for every bit Z_i, and we define the Hamiltonian corresponding to h_C as

$$H_{p,C}(|z_1\rangle|z_2\rangle...|z_n\rangle) = h_C(z_{iC}, z_{jC}, z_{kC})(|z_1\rangle|z_2\rangle...|z_n\rangle) \tag{85}$$

and the whole Hamiltonian is $H_P = \sum_C H_{P,C}$

satisfiability problem

In general, it is difficult to find the ground state of H_P. While we consider another easy constructing Hamiltonian H_B and its ground state also easy to find.

- the $H_{B,C}$ defined as a sum of one-qubit Hamiltonian, that is $H_{B,C} = H_B^{iC} + H_B^{jC} + H_B^{kC}$ where $H_B^i = (1 - \sigma_x)/2$
- the whole Hamiltonian will be $H_B = \sum_C H_{B,C}$
- the initial state H_B is $|x_1 = 0\rangle|x_2 = 0\rangle...|x_n = 0\rangle$

Now we will design a path to adiabatically connect the initial Hamiltonian H_B and H_P, that is,

$$H(t) = (1 - t/T)H_B + (t/T)H_P \tag{86}$$

if the evolution is slow enough, the ground state of H_B can adiabatically evolute to the ground state of H_P.

quantum adiabatic algorithm

- An easily constructible initial state, which is the ground state of H_B.
- A time-dependent Hamiltonian, $H(t)$, that is easily constructible from the given instance of the problem.
- An evolution time T determined by the gap of $H(t)$
- The final state $|\psi(T)\rangle$ that for T big enough will be (very nearly) the ground state of H_P.
- A measurement of $z_1, z_2, ..., z_n$ in the state $|\psi(T)\rangle$. The result of this measurement will be a satisfying assignment. If the formula has no satisfying assignment, the result will still minimize the number of violated clauses.

Grover problem

generally, the gap g_{min} is strongly dependent of the number of particles. The scaling of this dependence determine the complexity of the corresponding problem. We will discuss the algorithm by examples.

- The Grover problem: The single clause for this problem, h_G, which depends on all n bits with a unique (but unknown) satisfying assignment $w = w_1, w_2, ..., w_n$, the corresponding Hamiltonian is

$$H_P = I - |\omega\rangle\langle\omega| \tag{87}$$

- the total Hamiltonian $H(s)$ can be write as

$$H(s) = (1-s) \sum_{j=1,2,...,n} \frac{1}{2}(1-\sigma_x^j) + s(1-|0\rangle\langle 0|) \tag{88}$$

where $|\omega\rangle$ has locally unitary transform to $|0\rangle$ and the spectra is unchange.

Grover problem

- Since $H(s)$ and the initial state of $H(0)$ are symmetric under the interchange of any two qubits, we can just work in the $(n+1)-dimensional$ space of symmetrized states and define the symmetry operator $\vec{S}=(S_x,S_y,S_z)$

$$S_\alpha=\frac{1}{2}\sum_{j=1}^{n}\sigma_\sigma^j \tag{89}$$

- the state can be relabeled as m_z which range from $-n/2$ to $n/2$. And we can rewrite $H(s)$ as

$$H(s)=(1-s)[n/2-S_x]+s(1-|m_z=n/2\rangle\langle m_z=n/2|) \tag{90}$$

- the rest task is to estimate the gap scaling between the lowest two eigenvalues at some time s

$$H(s)|\psi\rangle = E|\psi\rangle \tag{91}$$

- operating the bra vector$\langle m_x = \frac{n}{2} - r|$ on the left and let $E = s + (1-s)\lambda$, we get

$$\frac{1-s}{s}\langle m_x = \frac{n}{2} - r|\psi\rangle = \frac{1}{r-\lambda}\langle m_x = \frac{n}{2} - r|m_z = \frac{n}{2}\rangle\langle m_z = \frac{n}{2}|\psi\rangle$$

- Multiply by $\langle m_z = \frac{n}{2}|m_x = \frac{n}{2} - r\rangle$ and sum over r to get

$$\frac{1-s}{s} = \sum_{r=0}^{n} \frac{1}{r - \lambda P_r} \tag{92}$$

where $P_r = |\langle m_z = \frac{n}{2}|m_x = \frac{n}{2} - r|^2 = \frac{1}{2^n}C_n^r$

Grover problem

- with the former eigenvalue equation,for certain s in $0 < s < 1$, the lowest two roots of the equation are $\lambda < 0$ and $0 < \lambda < 1$
- since we just consider the scaling of the gap, we consider the parameter s^* satisfy $\frac{1-s^*}{s^*} = \sum_{r=1}^{n} \frac{p_r}{r}$, then the eigenvalue equation will be

$$\frac{P_0}{\lambda} = \sum_{r=1}^{n} P_r \frac{\lambda}{r(r-\lambda)} \tag{93}$$

- let $\lambda = 2^{-n/2}u$,the former equation will be

$$\frac{1}{u} = \sum_{r=1}^{n} P_r \frac{u}{r(r-2^{-n/2}u)} \tag{94}$$

neglect the $2^{-n/2}u$ part to get $\frac{1}{u^2} = \sum_{r=1}^{n} \frac{P_r}{r^2}$

Grover problem

- with the former equation to get $\lambda = \pm(\sum_{r=1}^{n} \frac{p_r}{r^2})^{-1/2} 2^{-n/2}$, then the gap can be expressed as

$$g_m in \approx 2(1 - s^*)(\sum_{r=1}^{n} \frac{p_r}{r^2})^{-1/2} 2^{-n/2} \tag{95}$$

- So we can get the scaling of the gap

$$g_m in \approx 2.2^{-n/2} \tag{96}$$

Exact Cover problem

Exact Cover Problem is a NPC problem. Exact Cover is a restricted form of Satisfiability.

- An n-bit instance of Exact Cover is built up from clauses, each of which is a constraint imposed on the values of three of the bits, $z_i + z_j + z_k = 1$. An n-bit instance of Exact Cover is a list of triples (i, j, k) indicating which groups of three bits are involved in clauses. The problem is to determine whether or not there is some assignment of the n bit values that satisfies all of the clauses.

- H_B is the magnetic field in the $x-$direction,and all the qubits are initial at the x-direction

$$|\psi_g(0)\rangle = \frac{1}{2^{n/2}} \sum |z_1\rangle|z_2\rangle...|z_n\rangle \tag{97}$$

Exact Cover problem

- the Hamiltonian H_P defined as

$$H_P|z_1\rangle|z_2\rangle...|z_n\rangle = h(z_1, z_2, ..., z_n)|z_1\rangle|z_2\rangle...|z_n\rangle \tag{98}$$

where $h = \sum_C h_C$ and h_C is the cost function to violate the clause C.

- the time dependent Hamiltonian can be given by

$$H(t) = (1 - \frac{t}{T})H_B + \frac{t}{T}H_P \tag{99}$$

- we measure the state $|\psi(T)\rangle$ at time T, the probability to get the right answer (can be checked quickly) determined by parameter $|c_\omega|^2$
- the probability of success depend on the parameter T

Exact Cover problem

we consider this quantum algorithm by some random generate instances.

- scaling with n

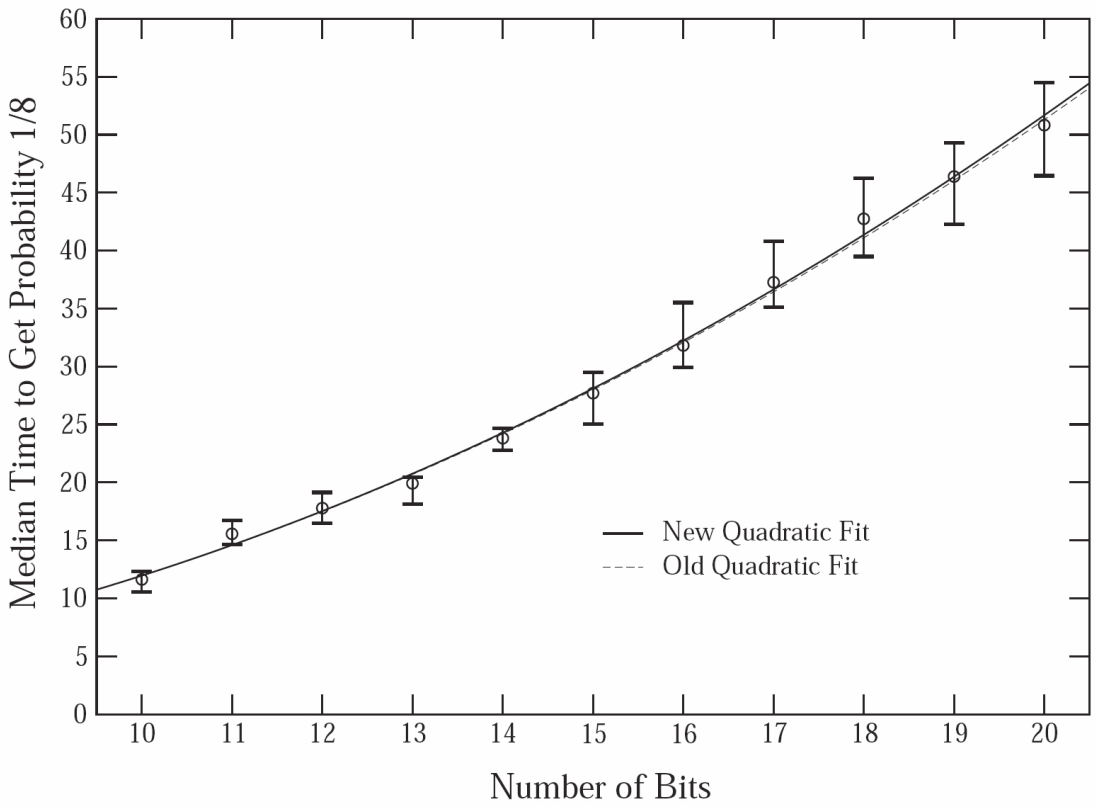


Figure 1: Each circle is the median time to achieve a success probability of 1/8 for 75 GUSA

Exact Cover problem

- phase transition

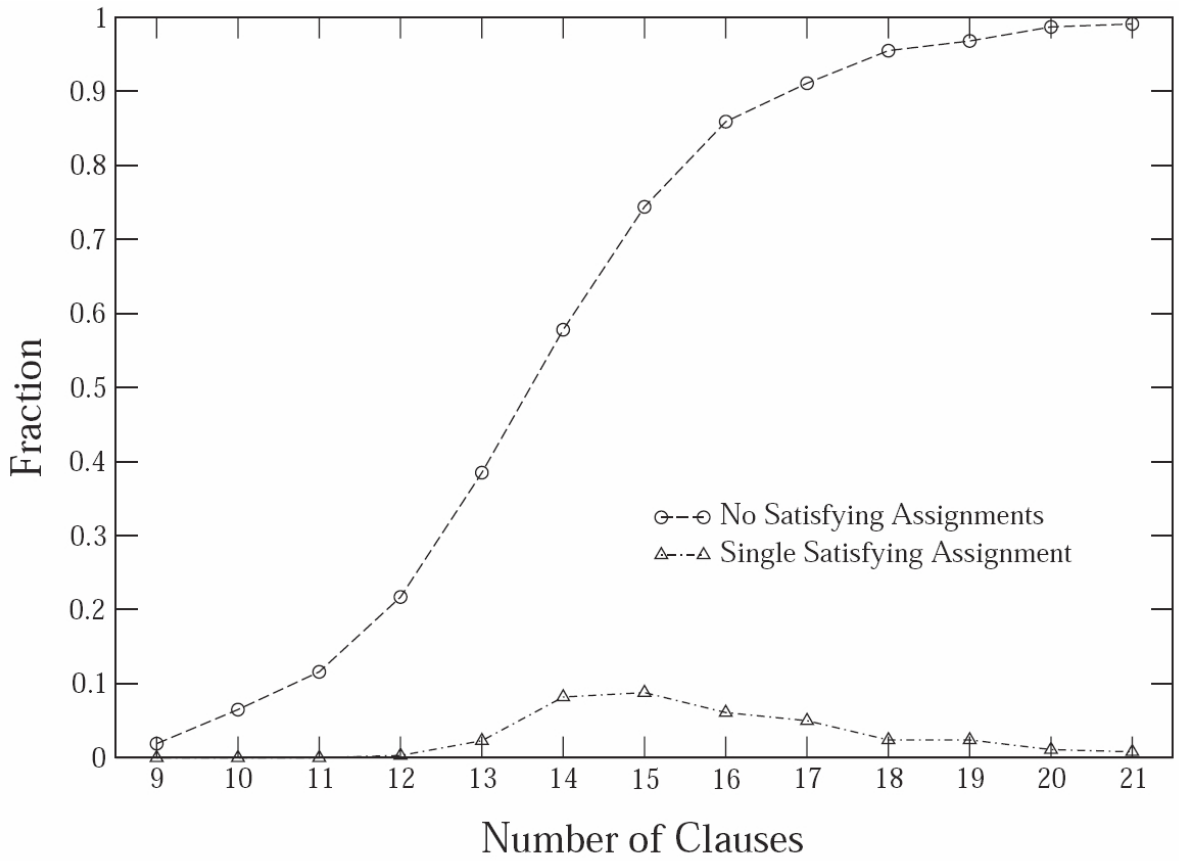


Figure 6: The circles give the fraction of instances with no satisfying assignment as a function of the number of clauses at 17 bits for Exact Cover. The triangles give the fraction of instances at each number of clauses that have a unique satisfying assignment.

Reference

- arxiv: 0909.4766
- arxiv: 0512159
- arxiv: 0201031
- arxiv: 0108048
- arxiv: 0001106

introduction

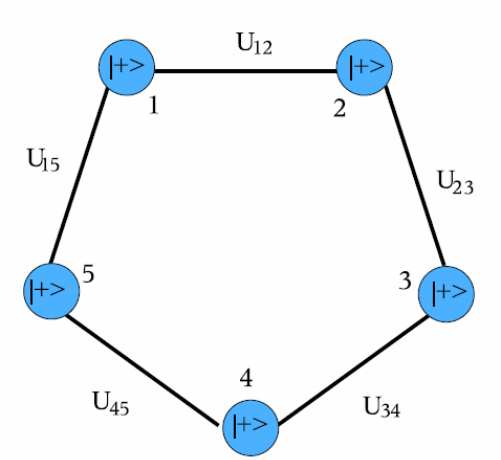
- one-way computer is another special quantum model of quantum computation, it show some key characters of quantum physics
- the power of one-way computer is the same as the former two models
- one-way computer have many advantages in discussion the fault tolerant computation
- it also convenient to use this model to investigate the relation between correlation and computation
- one-way computer just need one-qubit measurements after the state preparation

graph state

there are two ways to define a graph state for a given graph

- defined by operations which is suited to preparation the state
  - prepare all the qubits in the state $|+\rangle$
  - operate CZ on the qubits connected by edges of the graph, CZ is commute for each other

where $CZ = Diag\{1, 1, 1, -1\}$



graph state

- the other way to define a graph state is based on some stabilizers. For every vertices, we define a stabilizer for vertex a

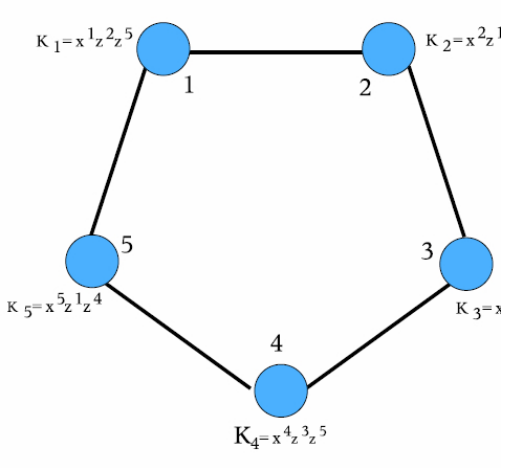
$$K_a = \sigma_a^x \Pi_{b \in N_a} \sigma_b^z \tag{100}$$

where N_a is the neighbor set of a all the stabilizers are commute,and the have the common eigenvector. all the stabilizers in the graph state is equal to 1. we define the Hamiltonian

$$H = -\sum_a K_a \tag{101}$$

the ground state of H is defined as the graph state $|G\rangle$.

character of graph state



- the stabilizers have a binary representation

$$(\mathbf{X}|\mathbf{Z}) = \left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right)$$

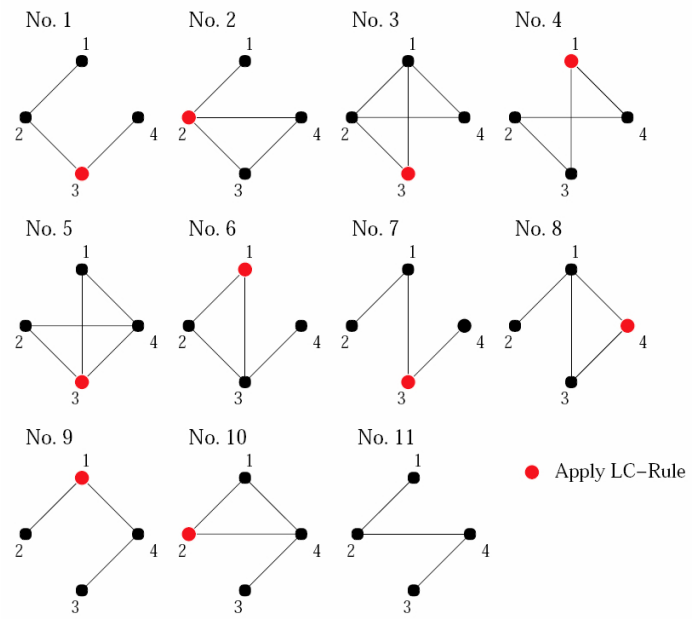
that is, $(X|Z) = (I|\Gamma)$ where γ is the adjacency matrix of graph G.

character of graph state

- all the state generate by $|W\rangle = \sigma_z^W |G\rangle$ will be a complete set of 2^n dimensional Hilbert space, where W is a $n - bit$ string which satisfies $K_a|W\rangle = (-1)^{W_a}|W\rangle$
- the density matrix of $|G\rangle$ is $\frac{1}{2^N}\sum_{\sigma\in S}\sigma$ where S is the set of the stabilizer. Let $A \in V$ be subset of vertices for a graph $G = (V, E)$ and B is the complement of A. The reduced state $\rho_G^A = tr_B(|G\rangle\langle G|)$ is given by $\rho_G^A = \frac{1}{2^{|A|}}\sum_{\sigma\in S_A}\sigma$ where σ supported by A.
- Pauli group: generated by $\{\pm i, \sigma_x, \sigma_y, \sigma_z\}$. Clifford group: transformations from Pauli group to Pauli group. If two graph states satisfy $|G'\rangle = C|G\rangle$ where C is an operator in clifford group, then these two graph states are equal under Clifford group
- any stabilizer state can be translated to a graph state by LC.

character of graph state

- By local complementation of a graph G at some vertex $a \in V$, one obtains an LC-equivalent graph state $|\tau_a(G)\rangle$,Furthermore, two graph states $|G\rangle$ and $|G\rangle$ are LC-equivalent iff the corresponding graphs are related by a sequence of local complementations.



where operate vertices is $3, 2, 3, 1, 3, 1, 3, 4, 1, 2$ respectively.

measurement of graph state

- A projective measurement of $\sigma_x, \sigma_y, or \sigma_z$ on the qubit associated with a vertex a in a graph G yields up to local unitaries $U^a_{i,}$, a new graph state $|G\rangle$ on the remaining vertices. The resulting graph G is

$$P^a_{z,\pm}|G\rangle = \frac{1}{\sqrt{2}}|z,\pm\rangle^a \bigotimes U^a_{z,\pm}|G-a\rangle \tag{102}$$

$$P^a_{y,\pm}|G\rangle = \frac{1}{\sqrt{2}}|y,\pm\rangle^a \bigotimes U^a_{y,\pm}|\tau_a(G)-a\rangle \tag{103}$$

$$P^a_{x,\pm}|G\rangle = \frac{1}{\sqrt{2}}|x,\pm\rangle^a \bigotimes U^a_{x,\pm}|\tau_{b_0}(\tau_a\tau_{b_0}(G)-a)\rangle \tag{104}$$

and the local unitaries are defined as

measurement of graph state

$$U^a_{z,+} = 1 \qquad U^a_{z,-} = \sigma_z^{N_a} \tag{105}$$

$$U^a_{y,+} = \sqrt{-i\sigma_z}^{N_a} \qquad U^a_{y,-} = \sqrt{+i\sigma_z}^{N_a} \tag{106}$$

$$U^a_{x,+} = \sqrt{+i\sigma_y}^{b_0}\, \sigma_z^{N_a - (N_{b_0} \bigvee b_0)} \tag{107}$$

$$U^a_{x,-} = \sqrt{-i\sigma_y}^{b_0}\, \sigma_z^{N_{b_0} - (N_a \bigvee a)} \tag{108}$$

That is the final graph can be obtained from the initial graph G by means of vertex deletion and local complementation:

- σ_z: deleting the vertex a from G;
- σ_y: inverting $G[N_a]$ and deleting a;
- σ_x: choosing any $b_0 \in N_a$, inverting $G[N_{b_0}]$, applying the rule for σ_y and finally inverting $\tilde{G}[N_{b_0}]$ again.

measurement of graph state

Furthermore, the commutation relation between Pauli measurement and Clifford operator can be given as

$$
\begin{aligned}
P_{x,\pm}\sigma_z &= \sigma_z P_{x,\mp},\\
P_{y,\pm}\sigma_z &= \sigma_z P_{y,\mp},\\
P_{z,\pm}\sigma_z &= \sigma_z P_{z,\pm},
\end{aligned}
$$

$$
\begin{aligned}
P_{x,\pm}(-i\sigma_z)^{1/2} &= (-i\sigma_z)^{1/2} P_{y,\mp},\\
P_{x,\pm}(i\sigma_y)^{1/2} &= (i\sigma_y)^{1/2} P_{z,\pm},\\
P_{x,\pm}(-i\sigma_y)^{1/2} &= (-i\sigma_y)^{1/2} P_{z,\pm},\\
P_{x,\pm}(i\sigma_z)^{1/2} &= (i\sigma_z)^{1/2} P_{y,\pm},
\end{aligned}
$$

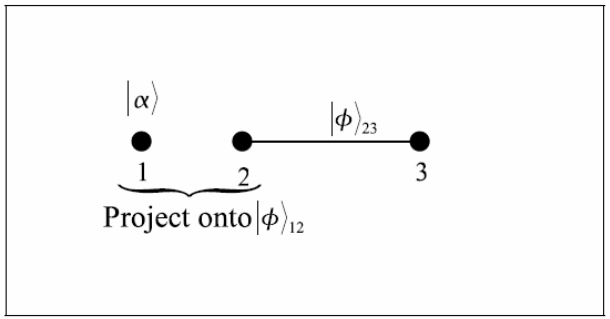
$$
\begin{aligned}
P_{y,\pm}(-i\sigma_z)^{1/2} &= (-i\sigma_z)^{1/2} P_{x,\pm},\\
P_{y,\pm}(i\sigma_y)^{1/2} &= (i\sigma_y)^{1/2} P_{y,\pm},\\
P_{y,\pm}(-i\sigma_y)^{1/2} &= (-i\sigma_y)^{1/2} P_{y,\pm},\\
P_{y,\pm}(i\sigma_z)^{1/2} &= (i\sigma_z)^{1/2} P_{x,\mp},
\end{aligned}
$$

$$
\begin{aligned}
P_{z,\pm}(-i\sigma_z)^{1/2} &= (-i\sigma_z)^{1/2} P_{z,\pm},\\
P_{z,\pm}(i\sigma_y)^{1/2} &= (i\sigma_y)^{1/2} P_{x,\pm},\\
P_{z,\pm}(-i\sigma_y)^{1/2} &= (-i\sigma_y)^{1/2} P_{x,\pm},\\
P_{z,\pm}(i\sigma_z)^{1/2} &= (i\sigma_z)^{1/2} P_{z,\pm},
\end{aligned}
$$

measure based computation

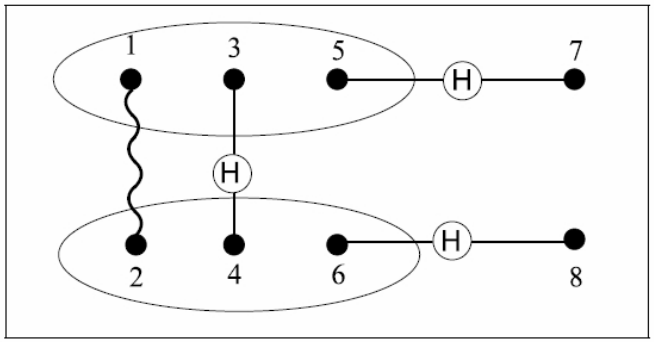
Based on the special entanglement graph state, the computation can be done on it by some one-qubit measurement.

- we can just prove that a universal gate set can be reached by one qubit measurement on graph state, here we consider the universal set including CZ gates and local one qubit unitary transformation.
- to make clear, we first consider the teleportation based computing
  - one qubit unitary transformation: The projection of $|\alpha\rangle_1|\phi\rangle_{23}$ onto $|\phi(U)_{12}$ results in the state $\frac{1}{d}U|\alpha\rangle_3$ at qubit 3, where $|\phi(U)\rangle = U^\dagger \bigotimes I|\phi\rangle$



measure based computation

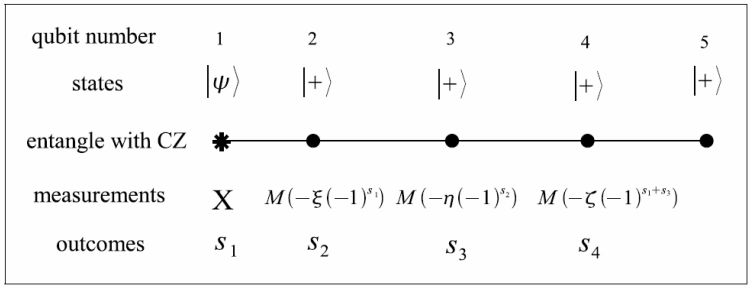
- two qubits CZ gate:



- The wiggly line connecting 12 denotes an input 2-qubit state $|\psi\rangle$. The lines labeled H denote the maximally entangled state $|H\rangle$.
- 3-qubit Bell measurement corresponding to the basis $|000\rangle \pm |111\rangle, |001\rangle \pm |110\rangle, |010\rangle \pm |101\rangle, |100\rangle \pm |011\rangle$.
- the measurement is performed on qubits 135 and 246 then the qubits 78 acquire the state $(P_i \bigotimes P_j)(H \bigotimes H)CZ|\psi\rangle$ where the Pauli operators P_i and P_j depend on the measurement outcomes.

measure based computation

- now we turn to the computation based on the graph state which only need one-qubit measurement which is different with the former teleportation based quantum computing.
  - one-qubit unitary transformation



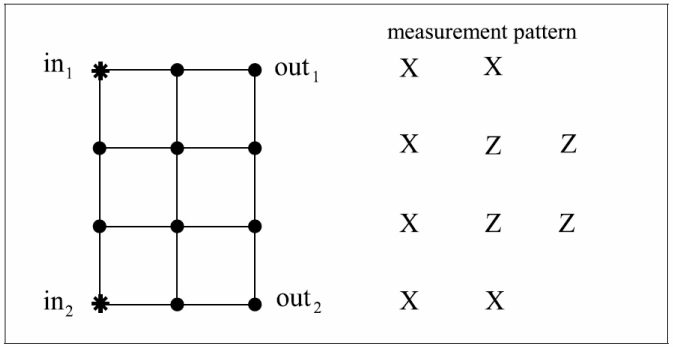
where ξ, η, and ζ are Euler angles for U, any U can be expressed as $U = R_x(\zeta)R_z(\eta)R_x(\xi)$.

- The leftmost qubit, denoted by a star, is set in state $|\psi\rangle$ and extended by a row of four $|+\rangle$ states denoted by dots. CZ operations are then applied, denoted by connecting lines.

measure based computation

- measurements are applied in the designated bases with outcomes s_i. Hence the measurements must be carried out adaptively from left to right.
- As a result of this process the rightmost (unmeasured) qubit is left in state $X^{s_2+s_4}Z^{s_1+s_3}U|\psi\rangle$

- two qubit CZ gate by one-qubit measurement on graph state



- The 2-qubit input state $|\psi_{in}\rangle$ is placed at sites labeled in_1 and in_2. Dots denote $|+\rangle$ states and connecting lines denote application of CZ for cluster state generation.

measure based computation

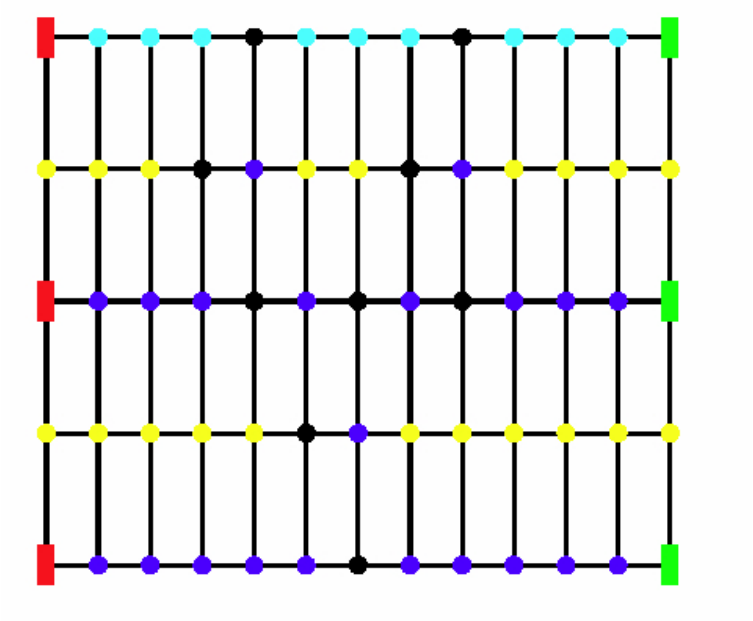
- If the measurement pattern shown at the right is applied at the sites, then only sites out_1 and out_2 remain unmeasured and contain $(P_1 \bigotimes P_2)CZ|\psi_{in}\rangle$ where $P_1 \bigotimes P_2$ is a Pauli operation that depends on the measurement outcomes.

so the measurement on certain configuration graph state can implement universal computation. To require the graph state can be used to universal computation, does it have some special structure of the graph?

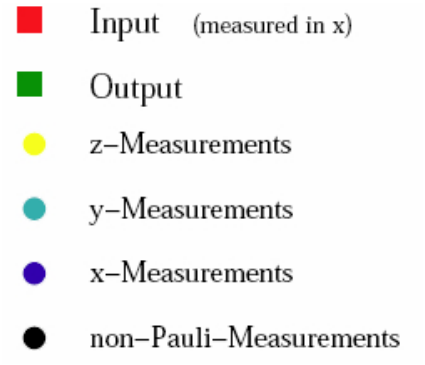
- one-dimension graph state is not universal for quantum computation. It can be simulated by classical computer efficiently.
- the computation power of a graph state maybe depend on the depth of its under graph. some popular lattice are universal.

measure based computation QFT

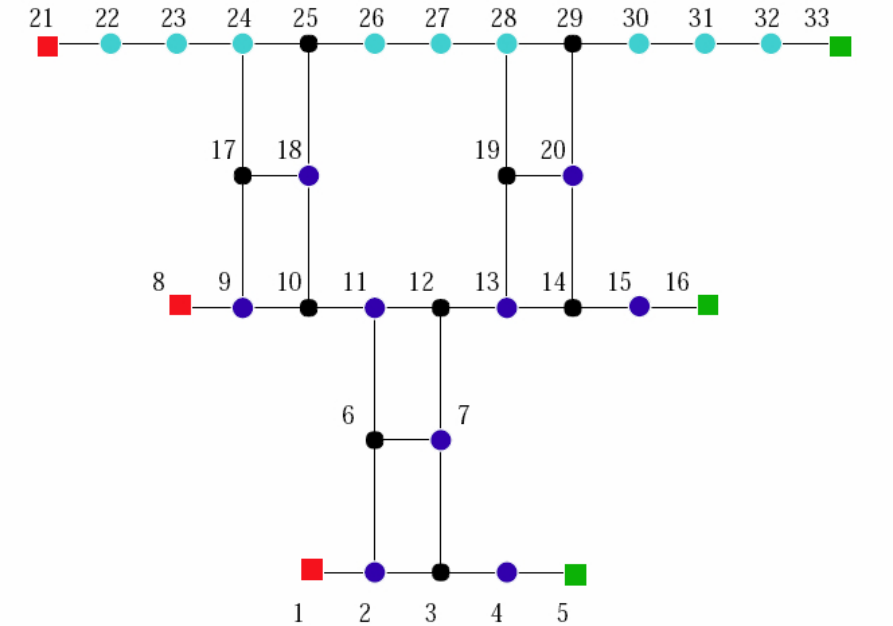
At last, we will give a one-qubit measurement on a square lattice to realize the simple 3-qubit QFT algorithm.



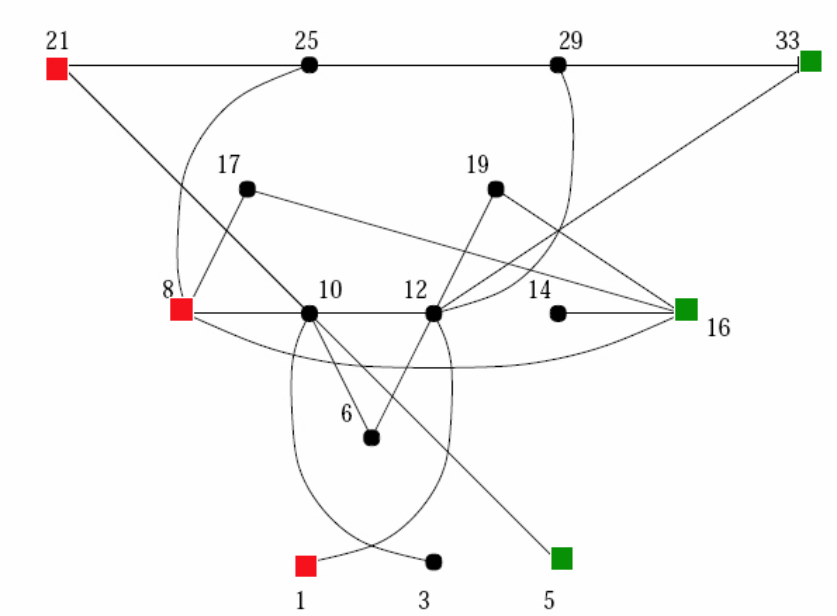
- this is the measure pattern on 13×5 lattice, the measurement like



- after the σ_z measurement, we get the following graph



- after all the Pauli measurement except the input qubits, we get the following graph



- arxiv: 0108067
- arxiv: 0504097
- arxiv: 0508124
- arxiv: 0602096
- arxiv: 0702116