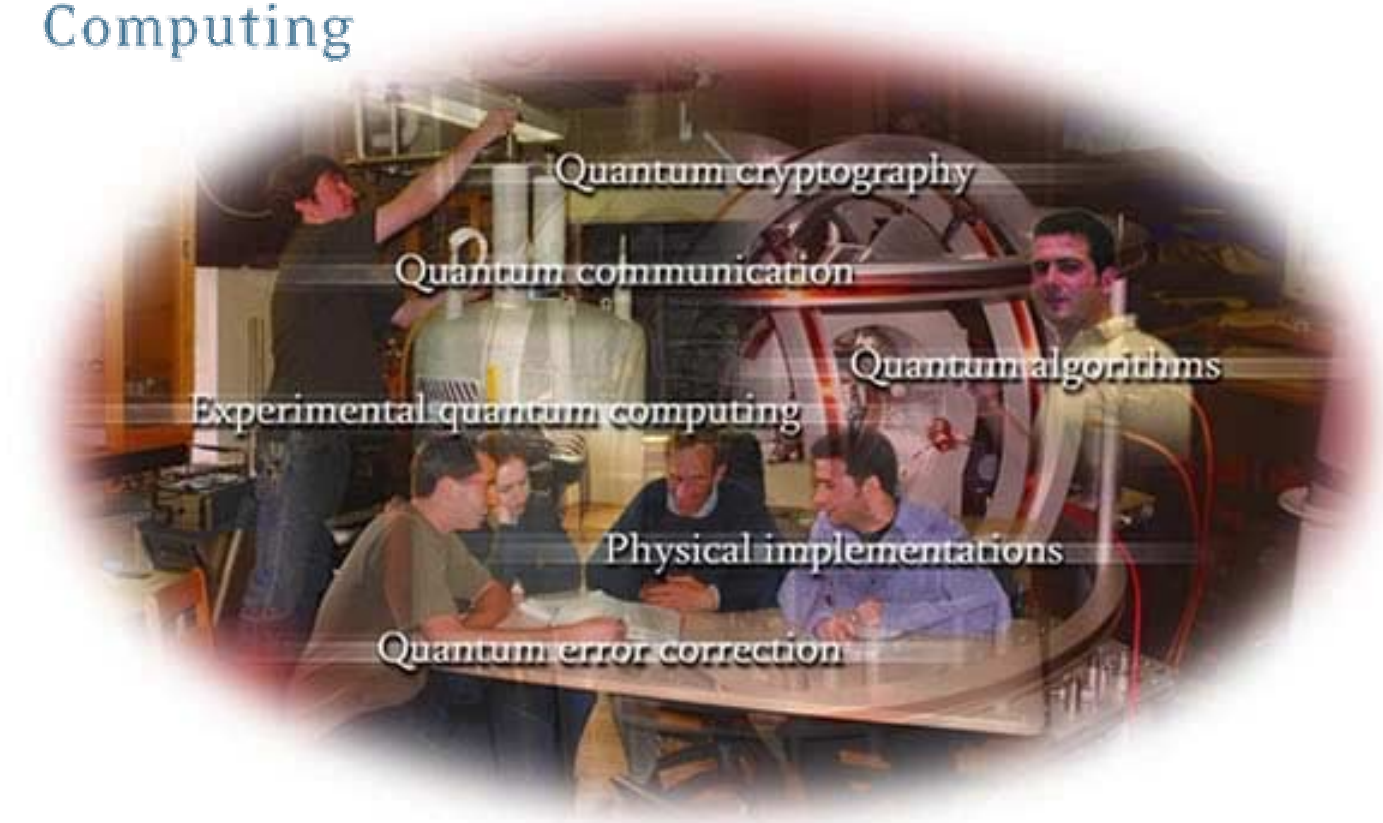
On the derandomization of quantum algorithms

Michele Mosca

Quantum Information Processing 2003

www.iqc.ca

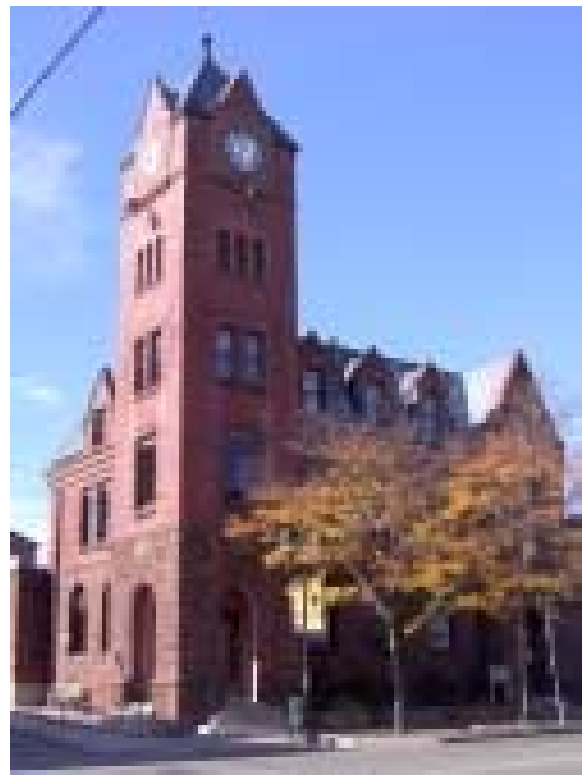
IQC Institute for Quantum Computing



PERIMETER INSTITUTE FOR THEORETICAL PHYSICS

Perimeter Institute is a community of theoretical physicists dedicated to investigating fundamental issues in theoretical physics.

www.perimeterinstitute.ca



PERIMETER INSTITUTE FOR THEORETICAL PHYSICS

What do I mean by "derandomization"?

- In classical probabilistic computing, "derandomization" is usually taken to mean the replacement of random coin flips with something deterministic. The consequence (and ultimate objective) is to turn probabilistic algorithms into deterministic ones.

- In the quantum setting, I take "derandomization" to mean turning probabilistic algorithms into deterministic or "exact" algorithms.

What algorithms do I consider?

I consider algorithms satisfying 3 properties:

- They succeed with probability at least ¼.
- We can efficiently and deterministically tell when they have succeeded.
- If the algorithm succeeds, we can compute the success probability exactly.

What was already known?

In Brassard and Hoyer's paper (quant-ph/9704027) on making Simon's algorithm exact, they introduce two essential ingredients:

- They tweek each iteration of algorithm to give a linearly independent vector orthogonal to hidden subgroup with probability exactly ½.
- They introduce a generalized version of amplitude amplification and get it to work exactly in the case that we know the success probability.

How does the amplitude amplification work?

- Consider any algorithm A (turned into a unitary quantum algorithm) that succeeds with probability $p = \sin^2(\theta)$

$$A|00\Lambda\ 0\rangle = \sin(\theta)|success\rangle + \cos(\theta)|failure\rangle$$

- Define $Q = -AU_0A^{-1}U_\chi$
- Then the algorithm $Q^k A|00\ \Lambda\ 0\rangle$ succeeds with probability

$$p = \sin^2((2k+1)\theta)$$

$$Q^k A|00\Lambda\ 0\rangle = \sin((2k+1)\theta)|success\rangle + \cos((2k+1)\theta)|failure\rangle$$

How does that work?

- Would like $$k = \frac{\pi}{4\theta} - \frac{1}{2}$$
- But this might not be an integer.
- Pick $$\tilde{k} = \left\lceil \frac{\pi}{4\theta} - \frac{1}{2} \right\rceil$$
- Høyer (quant-ph/0006031) and Brassard and Høyer suggest strategies that apply a modified quantum search iterate a total of $\tilde{k}$ times.
- Earlier work (BHMT, quant-ph/0005055) suggests another strategy that also works.

Alternative technique (Høyer)

- Pick $\tilde{k} = \left\lceil \frac{\pi}{4\theta} - \frac{1}{2} \right\rceil$
- Pick $\tilde{\theta} \leq \theta$ satisfying $\sin^2\left((2\tilde{k}+1)\tilde{\theta}\right) = 1$
- Prepare the qubit

$$\frac{\sin\left(\tilde{\theta}\right)}{\sin\left(\theta\right)}|1\rangle + \sqrt{1 - \frac{\sin^2\left(\tilde{\theta}\right)}{\sin^2\left(\theta\right)}}|0\rangle$$

i.e. Define

$$\tilde{A}\left|00\Lambda\ 0\right\rangle\left|0\right\rangle$$

$$=\left(\sin(\theta)\left|success\right\rangle+\cos(\theta)\left|failure\right\rangle\right)\left(\frac{\sin(\tilde{\theta})}{\sin(\theta)}\left|1\right\rangle+\sqrt{1-\frac{\sin^2(\tilde{\theta})}{\sin^2(\theta)}}\left|0\right\rangle\right)$$

$$=\begin{pmatrix}\sin(\tilde{\theta})\left|success\right\rangle\left|1\right\rangle+\sin(\theta)\sqrt{1-\frac{\sin^2(\tilde{\theta})}{\sin^2(\theta)}}\left|success\right\rangle\left|0\right\rangle\\ +\cos(\theta)\left|failure\right\rangle\left(\frac{\sin(\tilde{\theta})}{\sin(\theta)}\left|1\right\rangle+\sqrt{1-\frac{\sin^2(\tilde{\theta})}{\sin^2(\theta)}}\left|0\right\rangle\right)\end{pmatrix}$$

Alternative technique

$$\tilde{A}\left|00\Lambda\ 0\right\rangle\left|0\right\rangle = \left(\sin\left(\tilde{\theta}\right)\left|\overline{success}\right\rangle + \cos\left(\tilde{\theta}\right)\left|\overline{failure}\right\rangle\right)$$

- Define $\tilde{Q} = -\tilde{A}\,U_0\,\tilde{A}^{-1}U_{\tilde{\chi}}$

- Then

$$\tilde{Q}^k\,\tilde{A}\left|00\Lambda\ 0\right\rangle = \left|\overline{success}\right\rangle = \left|success\right\rangle\left|1\right\rangle$$

Why is it hard to derandomize some algorithms?

Two problems:

1. We don’t a priori know the success probability.
2. If we know the probability, we don’t know how to efficiently and exactly implement the necessary phase shifts or coin flips exactly within the computing model.

What can we do about Problem 1?

1. We don’t a priori know the success probability.

Possible solution: Massage the algorithm to work with a known probability.

Technique 1: Tweek the algorithm to make it easier to analyze the success probability.

Technique 2: Reject some of the successful outcomes with an appropriate probability.

Elaborate on Technique 2

- It suffices to be able to compute the success probability once we obtain a successful outcome

$$\tilde{A}\left|00\Lambda\ 0\right\rangle\left|0\right\rangle$$

$$= \begin{pmatrix} \sin(\theta)\left|success\right\rangle\left|\overline{\theta}\right\rangle\left(\frac{\sin(\tilde{\theta})}{\sin(\theta)}\left|1\right\rangle + \sqrt{1-\frac{\sin^2(\tilde{\theta})}{\sin^2(\theta)}}\left|0\right\rangle\right) \\ +\cos(\theta)\left|failure\right\rangle\left|0\right\rangle\left|0\right\rangle \end{pmatrix}$$

$$= \sin(\tilde{\theta})\left|\overline{success}\right\rangle + \cos(\tilde{\theta})\left|\overline{failure}\right\rangle$$

$\left|\overline{\theta}\right\rangle$ is a classical description of θ

$$\tilde{A}\left|00\Lambda\ 0\right\rangle\left|0\right\rangle = \left(\sin\left(\tilde{\theta}\right)\left|\overline{success}\right\rangle + \cos\left(\tilde{\theta}\right)\left|\overline{failure}\right\rangle\right)$$

- Define $\tilde{Q} = -\tilde{A}\,U_0\,\tilde{A}^{-1}U_{\tilde{\chi}}$

- Then

$$\tilde{Q}^k\,\tilde{A}\left|00\Lambda\ 0\right\rangle = \left|\overline{success}\right\rangle = \left|success\right\rangle\left|\overline{\theta}\right\rangle\left|1\right\rangle$$

What can we do about problem 2?

2. Once we know the probability, we don't know how to efficiently and exactly implement the necessary phase shifts or quantum coin flips exactly within the computing model.

- Should our computing model allow us to implement the required rotation or quantum coin flip? (Adleman, Demarrais and Huang, "Quantum Computability", define a notion of EQP relative to various quantum Turing machines.)

- *For now*, let's assume that we can deterministically compute the required one-qubit gate to m bits of precision in time polynomial in m and the input size.

Is this attempted derandomization technique better than standard Chernoff bound methods?

- Consider a black-box problem with an algorithm that on input size n makes N queries and uses T other gates and satisfies the three criteria
  - They succeed with probability at least ¼.
  - We can efficiently and deterministically tell when they have succeeded.
  - If the algorithm succeeds, we can compute the success probability exactly.
- Let's amplify the success probability to $1-\varepsilon$

Amplifying to success probability $1-\varepsilon$

New technique

3N queries

$$poly\left(\log\left(\frac{1}{\varepsilon}\right), n\right)T$$

other gates

Chernoff bound method

$$O\left(\log\left(\frac{1}{\varepsilon}\right)N\right)$$ queries

$$O\left(\log\left(\frac{1}{\varepsilon}\right)T\right)$$

other gates

Back to problem 2

- I claim that we should be allowed to use any one-qubit gate G for which we specify a deterministic classical Turing machine T_G that computes the amplitudes of the gate with m bits of precision in time polynomial in m and the length of the description of T_G.
- In other words, we can assume that we have an oracle that maps

$$\left|T_G\right\rangle\left|\psi\right\rangle \alpha \ \left|T_G\right\rangle G\left|\psi\right\rangle$$

Why allow such a gate?

- We need something like this to have a robust definition of exact quantum computation, i.e. one that does not depend on a arbitrary choice of universal gate set.

- We have already effectively seen this, but a finite number of times when defining a particular quantum computer (e.g. [BV97])

Why allow such an oracle?

- In the uniform family of circuits model, why not define a finite set of gates for each circuit produced?

- Why not allow it any time, as long as we charge appropriately?

Things we can do exactly and efficiently in this model

- The quantum Fourier transform for any product of cyclic groups.
- Finite Abelian HSPs
- Order-finding and integer factorization

Derandomizing Shor's factoring algorithm in this model

Recall the algorithm for splitting N into two non-trivial factors (CEMM version of Kitaev approach):

$$\sum_{a=1}^{N-1} \frac{1}{\sqrt{N-2}} |a\rangle$$ Pick a random a

$$\sum_{(a,N)=1} \frac{\sqrt{\varphi(N)}}{\sqrt{N}} |a\rangle|1\rangle$$ Test if gcd(a,N) > 1

$$+ \sum_{(a,N)>1} \frac{\sqrt{N-\varphi(N)}}{\sqrt{N}} |a\rangle|\gcd(a,N)\rangle$$

Derandomizing Shor's factoring algorithm in this model

For a fixed $|a\rangle$ where gcd(a,N)=1

$$\sum_{x=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |x\rangle |a^x \bmod N\rangle$$

exponentiate

$$= \sum_{k=0}^{r-1} \frac{1}{\sqrt{r}} \left(\sum_{x=0}^{2^n-1} \frac{1}{\sqrt{2^n}} e^{2\pi i \frac{k}{r} x} |x\rangle \right) |\Psi_{k/r}\rangle$$

Derandomizing Shor's factoring algorithm in this model

For a fixed $|a\rangle$ where gcd(a,N)=1

$$= \sum_{k=0}^{r-1} \frac{1}{\sqrt{r}} \left(\sum_{l=0}^{2^n-1} \alpha_l |l\rangle \right) \| \Psi_{k/r} \rangle$$ After the QFT

$$\frac{l}{2^n} \leq \frac{k}{r} < \frac{l+1}{2^n}$$

Can efficiently approximate this probability

$$\left|\alpha_l\right|^2 + \left|\alpha_{l+1}\right|^2$$

$$= \frac{\sin^2\left(\pi\left(\frac{2^n k}{r} - l\right)\right)}{2^n \sin^2\left(\pi\left(\frac{k}{r} - \frac{l}{2^n}\right)\right)} + \frac{\sin^2\left(\pi\left(\frac{2^n k}{r} - l - 1\right)\right)}{2^n \sin^2\left(\pi\left(\frac{k}{r} - \frac{l+1}{2^n}\right)\right)}$$

$$\geq \frac{8}{\pi^2}$$ (We will ultimately only accept these outcomes)

Can efficiently approximate this probability

After continued fractions, and test of denominator:

$$\sum_{k=0}^{r-1} \frac{1}{\sqrt{r}} \left(\sum_{y=0}^{2^n-1} \alpha_y \left| y \right\rangle \left| \tilde{k}, \tilde{r} \right\rangle \left| a^{\tilde{r}} \right\rangle \right) \left| \Psi_{k/r} \right\rangle$$

Attempted factorization:

$$\sum_{k=0}^{r-1} \frac{1}{\sqrt{r}} \left(\sum_{y=0}^{2^n-1} \alpha_y \left| y \right\rangle \left| \tilde{k}, \tilde{r} \right\rangle \left| a^{\tilde{r}} \right\rangle \left| \gcd\left(a^{\tilde{r}/2} - 1, N \right) \right\rangle \right) \left| \Psi_{k/r} \right\rangle$$

Can amplify success to a constant by doing two independent eigenvalue estimations

$$\left|y_1\right\rangle\left|\tilde{k}_1,\tilde{r}_1\right\rangle\left|\Psi_{k_1/r}\right\rangle\left|y_2\right\rangle\left|\tilde{k}_2,\tilde{r}_2\right\rangle\left|\Psi_{k_2/r}\right\rangle\left|\tilde{r}=lcm(\tilde{r}_1,\tilde{r}_2)\right\rangle$$

How can we compute the probability of success of this mess??

If we had the full prime factorization of

$$N, \varphi(N), \varphi\left(\varphi\left(N\right)\right), \Lambda\ , \varphi^{(t)}\left(N\right) = 1$$

$$t \le \log_4\left(N\right)$$

we could efficiently:

1. Redo the eigenvalue measurements exactly and accept only the two values $|l\rangle$ and $|l+1\rangle$ satisfying
$$\frac{l}{2^n} \le \frac{k}{r} < \frac{l+1}{2^n}$$
2. efficiently compute all the relevant probabilities needed to derandomize.

Overall strategy

Keep splitting N into factors until we have a prime power factorization. Then continue factoring $\varphi(N)$ and so on until we attempt a total of $2\lfloor \log^2 N \rfloor + 1$ splittings.

(Once we are down to 1, we can trivially "split" 1 into 1=1x1.)

Ideally

- Ideally, we would like each splitting to work with probability exactly ½ and we would keep only the computational paths that succeeded at least $\lfloor \log^2 N \rfloor + 1$ times (which is more than enough to get the full factorization of $N, \varphi(N), \Lambda$)
- Note that this event would occur with probability exactly ½.
- Along all such successful paths, we doctor the probabilities to give such a distribution.
- We can now easily derandomize the algorithm.

Summary

- Have introduced additional criteria under which we can quantumly derandomize algorithms

- With a reasonable computing model, this allows the derandomization of several existing BQP quantum algorithms