What to do with a small quantum computer?

Aram Harrow (MIT Physics)
simulating quantum mechanics is hard!

Implications:
1. Quantum mechanics is hard to simulate classically.
2. Quantum computers can do this simulation exponentially more efficiently.
3. Maybe they can do other things more efficiently too?
Motivation

Practical
• Social/economic value of computing
• End of Moore’s Law

Foundational
• Nature of information and computing
classical computers

\[ n \text{ bits} = 2^n \text{ states} \]

\[
\begin{array}{cccccccccccccccc}
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
\end{array}
\]

gates

\[ x \ \text{NAND} \ y \]
quantum computers

qubit = two-level system

nuclear spin

photon polarization

ion e- states

superconducting charge or flux

1 qubit = 2 dimensions

|ψ⟩ ∈ ℂ²

n qubits = 2ⁿ dimensions

|ψ⟩ ∈ ℂ²ⁿ

Schrödinger's eq:

\[
\frac{d|ψ⟩}{dt} = -iH(t)|ψ⟩
\]

quantum gates

I ⊗ ... ⊗ I ⊗ U ⊗ I ⊗ ... ⊗ I

(one- or two-qubit)
randomized computation

Application: high-dimensional integrals

$$\int \cdots \int dx_1 \cdots dx_n f(x_1, \ldots, x_n)$$

$$\begin{pmatrix}
p_{0000} \\
p_{0001} \\
\vdots \\
p_{1111}
\end{pmatrix} \in \mathbb{R}^{2^n}$$

FERMIAC
unstructured search

<table>
<thead>
<tr>
<th>x</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
</tr>
</tbody>
</table>

Given the ability to compute $f$, find $x$ such that $f(x)=1$.

**classical:** $O(N)$ time needed

**Grover’s algorithm:** (1996) $O(\sqrt{N})$ on quantum computer

**similar speedups** for maximizing, approximate counting, collisions, triangle finding, game-tree evaluation, backtracking algorithms ...
factoring

\[ 2139812390123891237129387 \times \, \, ? \times \, ? \]

n digits

Best classical algorithm: time \( O(\exp(n^{1/3})) \)

Shor’s algorithm: (1994)

time \( O(n^2) \) on a quantum computer

(similar for discrete log, elliptic curves, ...)

Best classical algorithm: time \( O(\exp(n^{1/3})) \)
linear systems of equations

\[ A \begin{pmatrix} \star & \star & \star \\ \star & \star & \star \\ \star & \star & \star \end{pmatrix} x = b \begin{pmatrix} \star \\ \star \\ \star \end{pmatrix} \]

- dimension \( N \)
- condition number \( \kappa \)
- up to \( s \) nonzero entries per row/column

**Classical:** \( O(Ns \kappa) \)

**Quantum:** \( O(\log(N)s \kappa) \)

- \( A \) computable on the fly input \( |b\rangle \) and output \( |x\rangle \)

**Applications:**
- ODEs
- PDEs
- regression
- machine learning
experimental QC

1-2 clean qubits

≈1000 noisy qubits

IBMs, Google, Microsoft, Intel, Alibaba, BBN, ...
Rigetti, psitech, ionQ, Quantum Circuits Inc, ...
proposals for ≈50-qubit universal QCs in 2-10 years using superconductors and ion traps

D-Wave
(1qbit, QxBranch, Lockheed Martin)
combinatorial optimization

Problem: minimize
\[ f(x_1, ..., x_n) = \sum \text{(terms involving a few variables)} \]

Applications:
• minimum energy of classical systems
• best way for a protein to fold
• best model in a family explaining observed data
• math proof with fewest errors
• ...

Difficulties:
• exponentially many local minima
• NP-complete even for good approximations
rest of the talk

classical optimization algorithms

adiabatic optimization

variational algorithms and quantum supremacy
classical local search

Local search
Start at point $x$
- Choose a neighbor $y$
- Move to $y$ if $f(x) > f(y)$
- Repeat

problem:
local minima
classical simulated annealing

goal: sample from Gibbs distribution

\[ \Pr[x] = e^{\frac{f(x)}{T}} / Z \]

annealing

gradually lower T

Metropolis algorithm

\[
\frac{R_{x \rightarrow y}}{R_{y \rightarrow x}} = e^{\frac{f(x) - f(y)}{T}}
\]

\( T=\infty \): unbiased random walk
\( T=0 \): local search
Walks on the hypercube

\{0,1\}^n = “n-dimensional hypercube”

adjacency matrix A

\[ A_{xy} = 1 \text{ if } (x,y) \text{ connected, } 0 \text{ if not} \]

\[
\begin{pmatrix}
000 & 011 & 010 & 011 & 100 & 101 & 110 & 111 \\
000 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
001 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
010 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
011 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
100 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
101 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
110 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
111 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
\end{pmatrix}
\]

Graph Laplacian: \( L = n \ I - A \)

continuous-time random walk: \( \frac{dp}{dt} = -Lp \Rightarrow \text{uniform distribution} \)

Simulated annealing: \[
\frac{dp}{dt} = -\text{diag}(e^{f/2T}) \cdot L \cdot \text{diag}(e^{-f/2T})p
\]
overview

classical optimization algorithms

adiabatic optimization

variational algorithms and quantum supremacy
quantum mechanics to the rescue

Diffusion equation:
\[ \frac{d}{dt} p(\vec{r}, t) = -L p(\vec{r}, t) \]
\[ L = \nabla^2 \]

Schrödinger equation:
\[ \frac{d}{dt} \psi(\vec{r}, t) = i L \psi(\vec{r}, t) \]
math-physics dictionary

eigenvectors of $H$ ↔ quantum states

eigenvalues of $H$ ↔ energies

eigenvector with lowest eigenvalue ↔ ground state

Schrödinger’s eq:
$$\frac{d|\psi\rangle}{dt} = -iH(t)|\psi\rangle$$

energy = frequency
adiabatic algorithm

Adiabatic theorem:
Slowly changing $H(t)$
$\rightarrow \approx$ stay in ground state

diagram showing $H_{\text{initial}}$ and $H_{\text{final}}$ with Schrödinger's equation:
$$\frac{d|\psi\rangle}{dt} = -iH(t)|\psi\rangle$$

$\langle \text{ground state (t=0)} | \rightarrow | \text{ground state (t=T_{final})} \rangle$

easy to prepare

desired output

minimum gap
adiabatic optimization

\[ H(t) = \left( 1 - \frac{t}{T_{\text{final}}} \right) L + \frac{t}{T_{\text{final}}} \text{diag}(f) \]

\[ \text{Schrödinger's eq: } \frac{d|\psi\rangle}{dt} = -iH(t)|\psi\rangle \]

\( H_{\text{initial}} = L \)

\( H_{\text{final}} = \text{diag}(f) \)

\( |\text{ground state (t=0)}\rangle \rightarrow |\text{ground state (t=T}_{\text{final}}\rangle \)

easy to prepare

desired output

\[ \frac{1}{\sqrt{2^n}} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \]

\[ \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \]

arg \( \min_x f(x) \)
tunneling and optimization

Analogue for adiabatic optimization

$f(|z|)\newline
\begin{align*}
|z| &= \text{# 1's in string } z
\end{align*}$
adiabatic tunneling beats simulated annealing

Adiabatic optimization fast if barrier area $\leq n^{1/2}$

Simulated Annealing exponential slowdown
quantum advantage in tunneling?

“...classical algorithms can only `walk over this landscape’. Quantum computers can **tunnel** through the landscape making it faster to find the **lowest point**. The D-Wave processor considers all the possibilities simultaneously to determine the lowest energy...”

from http://www.dwavesys.com/quantum-computing
ground states of stoquastic Hamiltonians

\[ H(t) = \left( 1 - \frac{t}{T_{\text{final}}} \right) L + \frac{t}{T_{\text{final}}} \text{diag}(f) \]

\[
\begin{pmatrix}
\leq 0 & \leq 0 & \cdots \\
\leq 0 & \leq 0 & \\
\leq 0 & & \\
\vdots & & \\
\end{pmatrix}
\]

“sign-problem free”

“stoquastic”

⇒ ground state \( \psi \) has only nonnegative entries

⇒ non-universal form of quantum computing

Quantum Enhanced Optimization
Quantum Monte Carlo (QMC) (actually a classical algorithm)

Stoquastic

\[ H = f + \Gamma \]

Vertical bonds:

ferromagnetic energy \( \approx \log(m/\Gamma) \)

i.e. disagree prob \( O(\Gamma/m) \)

Horizontal bonds:

= \( f / m \)
QMC vs barriers

**Thm:** For barrier sizes/shapes where the adiabatic algorithm tunnels through rapidly then quantum Monte Carlo does too.

**Proof idea:**

Only a few rows on the barrier at once.
simulated annealing vs quantum adiabatic optimization vs quantum Monte Carlo

Annealing: $T = 1.50$

Adiabatic: Transverse Field = 3.00
implications

• Evidence against tunneling as a route to quantum speedup.

• Routes to quantum advantage?
  • Non-stoquastic adiabatic optimization
  • Tasks where quantum Monte Carlo fails
overview

classical optimization algorithms

adiabatic optimization

variational algorithms and quantum supremacy
variational quantum algorithms

Theorem: “Quantum Supremacy”
Outputs of quantum algorithm are hard to simulate classically.
things I don’t know

- Why are there quantum speedups?
  - Exponentially large dimension not enough.
  - Interference needed too.
- What good is a 100-qubit quantum computer?
- Stoquastic adiabatic evolution: quantum supremacy or efficient classical simulation?
deleted scenes
Use local moves (Glauber or Metropolis) to generate samples from $\pi(z_1, ..., z_L)$. Run-time/accuracy tradeoff unknown in general.

2. Use sampling-to-counting equivalence to estimate $Z$ or $\langle O \rangle = \text{tr}[O \ e^{-\beta H}]/Z$.

Problem reduces to bounding mixing time (equiv. gap) of a classical Markov chain.
The Markov chain

| m | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| n | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

weight(z) = \( \exp(-\beta (f(z_1) + \ldots + f(z_L)) / m) \times (\beta \Gamma / m)^{\# \text{ vertical jumps}} \)

rapidly mixing?
the path measure

random walk $z_1, ..., z_m$ on hypercube $\{0,1\}^n$
  - conditioned to return ($z_{m+1} = z_1$)
  - alternatively can use open boundary conditions.
  - typically $\approx \beta n$ total jumps

Suppose that $f(z)$ depends only on Hamming weight $|z|$.
  - look only at Hamming weight: $\{0,1\}^n \rightarrow \{0,1, ..., n\}$.
  - take $n \rightarrow \infty$ and $\{0,1, ..., n\} \rightarrow [0,1]$.
  - Brownian motion, or with closed B.C., “Brownian bridge”

with local $Z$ fields $\rightarrow$
  Brownian motion with drift
  “Ornstein-Uhlenbeck bridge”
  $dx(t) = \theta (\mu - x(t)) dt + \sigma dB(t)$
  $\theta = $ drift, $\mu = $ mean, $\sigma = $ diffusion

[see also JSIBMTN 1603.01293]
Local times of Brownian motion

**Local time:** $L^x(t) = \text{amount of time Brownian motion } B(t) \text{ spends at point } x.$

**Lévy's theorem:** \{L^0(t): t \geq 0\} and \{S(t): t \geq 0\} have the same distribution, where $S(t) = \sup_{0 \leq s \leq t} B(s)$.

In fact, $(S-B, S) =^d (|B|, L^0)$
Additionally, $S =^d |B|$. 
**Local times of Brownian motion**

**Local time:** $L^x(t)$ = amount of time Brownian motion $B([0,t])$ spends at point $x$.

**Lévy's theorem:** $(S-B, S) =^d (|B|, L^0)$

**Proof:** consider discrete walk: $W(k) = X(1) + \ldots + X(k)$ with $X(t) = \pm 1$. Let $M(k) = \max(W(0), \ldots, W(k))$.

$$Y(t) = M(t) - W(t) \approx^d |W(t)|$$

$M = \#$ of times $M-W$ remains at $0 \approx^d L^0$

Figure adapted from *Brownian Motion* by Mörters and Peres.
Hamming weight + spike

\[ f(|z|) \]

width \( n^a \)

height \( n^b \)

\[ \exp\left(-n^{\frac{a}{2}} + \frac{b-1}{2}\right) \]

\[ \Omega(1) \]

\[ n^{1/2-a-b} \]
QMC and tunnelling

s\text{pike (width } n^a \text{, height } n^b\text{)}

\begin{align*}
\text{ST} &= \text{normalized spike time} \\
&\approx_d |N(0, n^{a-1/2})|
\end{align*}

proof using either Lévy’s thm or quantum-classical correspondence.

\text{Feynman-Kac thm:}
\Pr[\text{path | spike}] = \exp(-\beta \text{ ST } n^b) \Pr[\text{path | no spike}]

\Rightarrow \text{if } a+b<1/2 \text{ then typical paths don’t notice the spike.}
instantons on the cheap

spike
(width \(n^a\), height \(n^b\))

\[ a < 1/2 \]
\[ 2a + b < 1 \]

steps to traverse spike \(\approx n^{2a}\)

\[ \text{min ST} = n^{2a} / \beta \Gamma n \]

Feynman-Kac \(\rightarrow\)
prob reduced by \(\approx \exp(-n^{2a+b-1})\)

\[ \therefore 2a + b \leq 1 \] is the threshold to cross the spike once.

cf JSIBMTN’16
canonical paths

Given Markov chain $P(x,y)$ with stationary distribution $\pi(x)$ and $Q(x,y) = P(x,y) \pi(y) = Q(y,x)$. TFAE:

- $P$ has a $\geq 1/poly(n)$ gap between the top two eigenvalues
- The conductance $\Phi$ is $\geq 1/poly(n)$. $\Phi = \min_S Q(S, S^c) / \pi(S) \pi(S^c)$
- For any $x,y$ there exists a path $\gamma_{xy}$ from $x \to y$ routing $\pi(x) \pi(y)$ units of flow such that each edge $e$ has load $\leq poly(n) Q(e)$. ("canonical paths/flows")

Heuristics analyze some plausible cut. Proofs analyze all cuts or construct paths.
1-d canonical path

<table>
<thead>
<tr>
<th>x_{1,1}</th>
<th>x_{2,1}</th>
<th>x_{3,1}</th>
<th>x_{4,1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_{1,2}</td>
<td>x_{2,2}</td>
<td>x_{3,2}</td>
<td>x_{4,2}</td>
</tr>
<tr>
<td>x_{1,3}</td>
<td>x_{2,3}</td>
<td>x_{3,3}</td>
<td>x_{4,3}</td>
</tr>
<tr>
<td>x_{1,4}</td>
<td>x_{2,4}</td>
<td>x_{3,4}</td>
<td>x_{4,4}</td>
</tr>
<tr>
<td>x_{1,5}</td>
<td>x_{2,5}</td>
<td>x_{3,5}</td>
<td>x_{4,5}</td>
</tr>
<tr>
<td>x_{1,6}</td>
<td>x_{2,6}</td>
<td>x_{3,6}</td>
<td>x_{4,6}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>y_{1,1}</th>
<th>y_{2,1}</th>
<th>x_{3,1}</th>
<th>x_{4,1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>y_{1,2}</td>
<td>y_{2,2}</td>
<td>x_{3,2}</td>
<td>x_{4,2}</td>
</tr>
<tr>
<td>y_{1,3}</td>
<td>y_{2,3}</td>
<td>x_{3,3}</td>
<td>x_{4,3}</td>
</tr>
<tr>
<td>y_{1,4}</td>
<td>y_{2,4}</td>
<td>x_{3,4}</td>
<td>x_{4,4}</td>
</tr>
<tr>
<td>y_{1,5}</td>
<td>x_{2,5}</td>
<td>x_{3,5}</td>
<td>x_{4,5}</td>
</tr>
<tr>
<td>y_{1,6}</td>
<td>x_{2,6}</td>
<td>x_{3,6}</td>
<td>x_{4,6}</td>
</tr>
</tbody>
</table>

... [Diagram of a transition matrix with highlighted transitions]

Energy penalty:
- ≤ 2 new jumps
- ≤ 1 term from $H_D$ (L bonds each with weight 1/L.)