Quantum Algorithms for Applied Mathematics

Stephen Jordan

Feb 17, 2015

JOINT CENTER FOR QUANTUM INFORMATION AND COMPUTER SCIENCE

UNIVERSITY OF MARYLAND

Quantum Algorithms

- Linear Algebra
- Integrals & Sums
- Optimization
- Simulation of Physical Systems
- Graph theory
- Number Theory
- Search and Boolean Formula Evaluation
- Topological Invariants

math.nist.gov/quantum/zoo/

Quantum Algorithms

- Linear Algebra
- Integrals & Sums
- Optimization

} This talk

- Simulation of Physical Systems

} Alán's talk

- Graph theory
- Number Theory
- Search and Boolean Formula Evaluation
- Topological Invariants

math.nist.gov/quantum/zoo/

Linear Algebra

Problem 1: given NxN matrix A and vector b, solve Ax=b.

- Classical complexity:
  - General Matrix:
    - Naive algorithm: $O(N^3)$
    - Best asymptotic complexity: $O(N^{2.373})$ (not practical)
  - Sparse Matrix: $O(N)$
- Quantum speedup:
  - Can partially characterize x in $O(\log N)$ time.

    [Harrow, Hassidim, Lloyd 2009]

Input Format

Q. If M is an NxN matrix, how can we compute anything about it in O(log N) time?

- We need N^2 time just to read the entries.
- For nonsingular sparse M we need at least N time.

A. If the entries of M are computable on the fly, then we can call this subroutine as needed.

HHL Quantum Algorithm

- Physics:

$$\vec{\psi}(t) = U(t)\vec{\psi}(0)$$

$$U(t) = e^{-iHt}$$

Hamiltonians H arising in nature are structured and sparse.

- Quantum algorithms for simulation [Alán's talk]
  - For physical H on n qubits, $U(t) = e^{-iHt}$ can usually be implemented with poly(n,t) quantum gates.
  - Matrix computation by quantum computation...

HHL Quantum Algorithm

Harrow Hasidim & Lloyd showed this primitive:

$$\vec{\psi} \rightarrow e^{-iHt}\vec{\psi}$$

can be used to construct a state:

$$\vec{\psi'} \propto H^{-1}\vec{\psi}$$

Problem 1: given NxN matrix A and vector b, solve Ax=b.

$$\vec{\psi} = \begin{bmatrix} b_1 \\ \vdots \\ b_{2^n} \end{bmatrix} \qquad H = A \qquad \xrightarrow{\text{make}} \qquad \vec{\psi'} \propto \begin{bmatrix} x_1 \\ \vdots \\ x_{2^n} \end{bmatrix}$$

Applications of HHL Algorithm

Problem 1: given NxN matrix A and vector b, solve Ax=b.

- Find instances of problem 1 such that:
  - A corresponds to efficiently simulatable Hamiltonian (e.g. sparse)
  - b corresponds to efficiently constructable quantum state vector
  - We only want to learn $x^T M x$ for some M that is also an efficiently simulatable Hamiltonian.

Applications of HHL Algorithm

Several applications have been proposed for the HHL algorithm and its variants*:

- Solving linear differential equations
  [Berry, 2010]
- Least-squares curvefitting
  [Wiebe, Braun, Lloyd, 2012]
- Machine learning
  [Lloyd, Mohseni, Rebentrost, 2013]
  [Lloyd, Garnerone, Zanardi, 2014]
- Matrix inversion in logarithmic space
  [Ta-Shma, 2013]

*dependence on condition number of M has been improved from k^2 to $k \log^3 k$ in [Ambainis, 2010]

Applications of HHL Algorithm

Several applications have been proposed for the HHL algorithm and its variants*:

Can you find more?

*dependence on condition number of M has been improved from k^2 to $k \log^3 k$ in [Ambainis, 2010]

Integrals & Sums

Problem 2: given black-box access to

$f : \{1, \ldots, N\} \to \mathbb{Z}$ approximate $\sum_{x=1}^{N} f(x)$ within $\pm\epsilon$.

- Classical randomized sampling: $\frac{1}{\epsilon^2}$
- Quantum amplitude amplification: $\frac{1}{\epsilon}$
  [Brassard, Hoyer, Tapp, 1998] [Mosca, 1998]
  - This is optimal. [Nayak, Wu, 1999]
  - Generalizes to integrals. [Novach, 2000]

Optimization

Problem 3: given some objective function $f : S \to \mathbb{R}$ find $x \in S$ minimizing f(x).

- Practical Examples are Everywhere:
  - Discrete: Given a list of N cities, find the shortest route visiting all of them.
  - Continuous: Given a parametrized family of nosecone shapes, find the one with the lowest coefficient of drag.
- I will focus on simplified examples, because analysis of quantum optimization is hard!

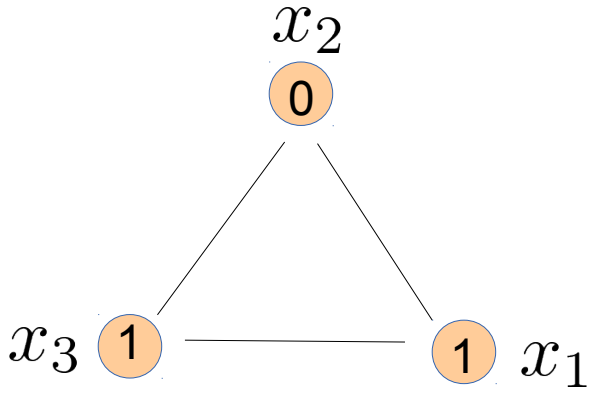
Optimization

Problem 3: given some objective function $f : S \to \mathbb{R}$ find $x \in S$ minimizing f(x).

Discrete example: Max E2LIN2

Given a list of linear constraints mod 2 on n variables find the assigment of variables violating as few as possible.

$$x_1 \oplus x_2 = 1$$
$$x_2 \oplus x_3 = 1$$
$$x_3 \oplus x_1 = 1$$



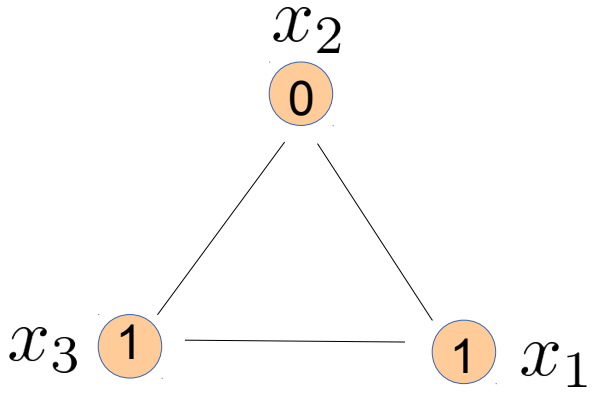
Optimization

Problem 3: given some objective function $f : S \to \mathbb{R}$ find $x \in S$ minimizing f(x).

Discrete example: Max E2LIN2

Given a list of linear constraints mod 2 on n variables find the assigment of variables violating as few as possible.

$$x_1 \oplus x_2 = 1$$
$$x_2 \oplus x_3 = 1$$
$$x_3 \oplus x_1 = 1$$



Optimal: violate 1.

Optimization

Problem 3: given some objective function $f : S \to \mathbb{R}$ find $x \in S$ minimizing f(x).

Discrete example: Max E2LIN2

Given a list of linear constraints mod 2 on n variables find the assignment of variables violating as few as possible.

- Search space is exponentially large: $|S| = 2^n$
- Can simplify analysis by only counting queries.

Complexity of Optimization

Theorem: If f is completely unstructured

(e.g. random ranking $f : \{0,1\}^n \rightarrow \{1,2,\ldots,2^n\}$)

then the optimal strategy is:

Classical: Brute search $O(2^n)$

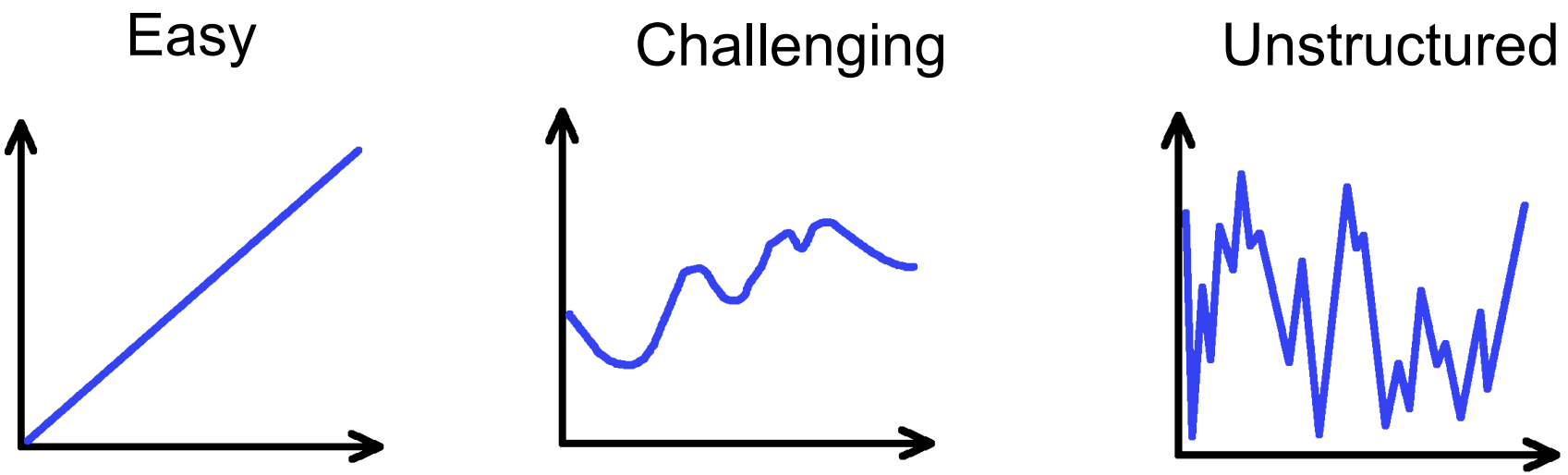
Quantum: Grover's algorithm $O(\sqrt{2^n})$

[Nayak, Wu, 1998]

Complexity of Optimization

In practice, the search space usually has some topology relative to which the objective function is smooth or structured.

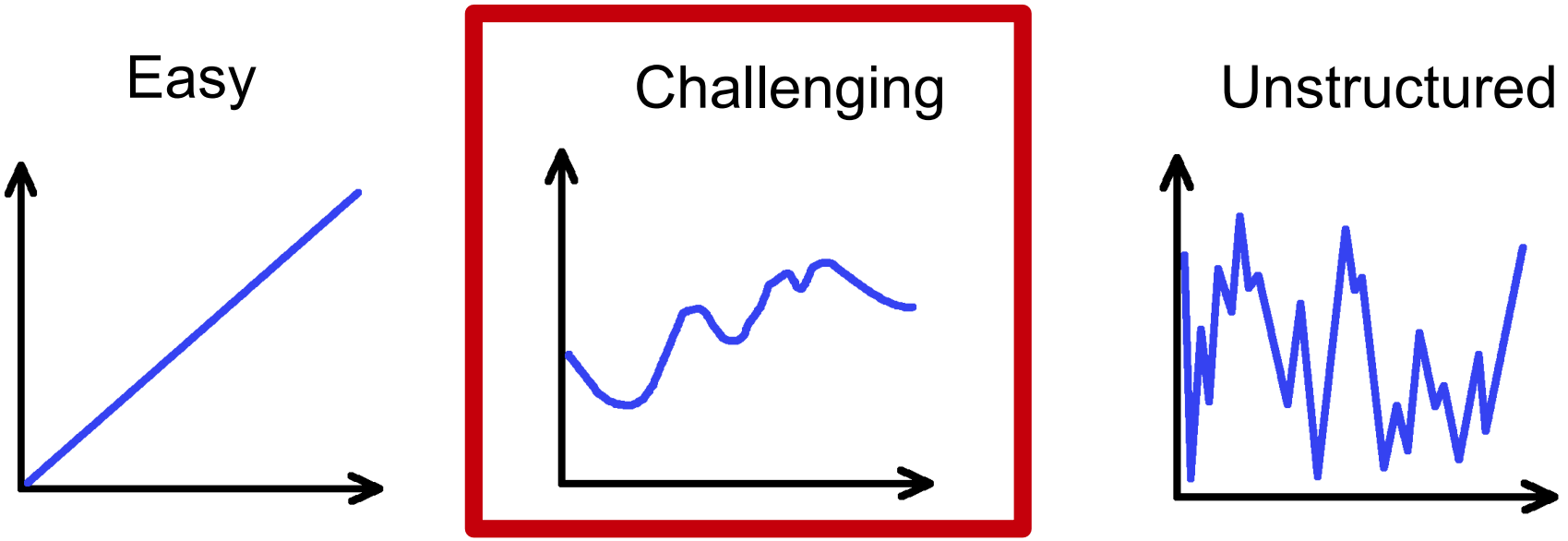
Example: $S = [0, 1] \subset \mathbb{R}$



Complexity of Optimization

In practice, the search space usually has some topology relative to which the objective function is smooth or structured.

Example: $S = [0, 1] \subset \mathbb{R}$



Complexity of Optimization

In practice, the search space usually has some topology relative to which the objective function is smooth or structured.

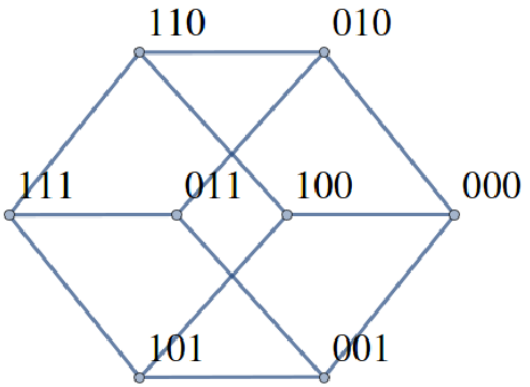
Example: Max E2LIN2 $S = \{0,1\}^n$

$$x_1 \oplus x_2 = 1$$
$$x_2 \oplus x_3 = 1$$
$$x_3 \oplus x_1 = 1$$

If $x, y \in \{0,1\}^n$ differ by flipping one bit, we can change f by only the number of constraints in which that bit appears.

Gradient Descent

- Flipping one bit is the natural local move.
- We can think of E3LIN2 as searching a space that has the topology of the hypercube graph.



- No local minima: we can use gradient descent.
- Otherwise we need some procedure to escape local minima.

Simulated Annealing

- We can modify gradient descent by accepting "uphill" moves with some nonzero probability so that we don't get stuck in local minima.

- Simulated annealing:

$$p_{\text{accept}} = e^{-\Delta f/T}$$ (Metropolis Rule)

Lower T slowly

Quasi-static probability distribution is $p(x) \sim e^{-f(x)/T}$

$T = \infty$ uniform

$T = 0$ probability 1 on minimum.

Adiabatic Quantum Computation

- In quantum mechanics, the energy of a system is determined by the Hamiltonian.

$$H\vec{\psi} = E\vec{\psi} \quad \Longleftrightarrow \quad \text{state } \vec{\psi} \text{ has energy E}$$

- The dynamics is also determined by the Hamiltonian:

$$\frac{d}{dt}\vec{\psi} = -iH(t)\vec{\psi}$$

- **Adiabatic Theorem**: if we start with the ground state and vary the Hamiltonian sufficiently slowly, the system tracks the ground state.

Adiabatic Quantum Computation

Idea [Farhi, Goldstone, Gutman, 1999]:

- Choose H(s) such that:
  - H(0) has ground state that is easy to prepare
  - H(1) has ground state that encodes solution to problem
- Slowly vary s from 0 to 1
- By adiabatic theorem, we obtain the solution if we go slowly enough.
- Quantitatively runtime is* $\widetilde{O}(1/\gamma^2)$, where γ is minimum gap between lowest and second-lowest eigenvalues of H(s).

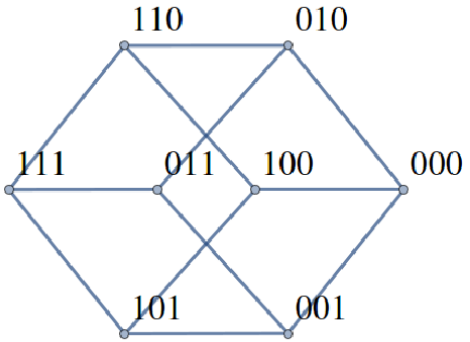
*See: [Elgart & Hagedorn, 2012]

Adiabatic Optimization

- Suppose $H(1) = \begin{bmatrix} f(0) & & & \\ & f(1) & & \\ & & \ddots & \\ & & & f(N) \end{bmatrix}$

  then ground state of H(1) minimizes f.

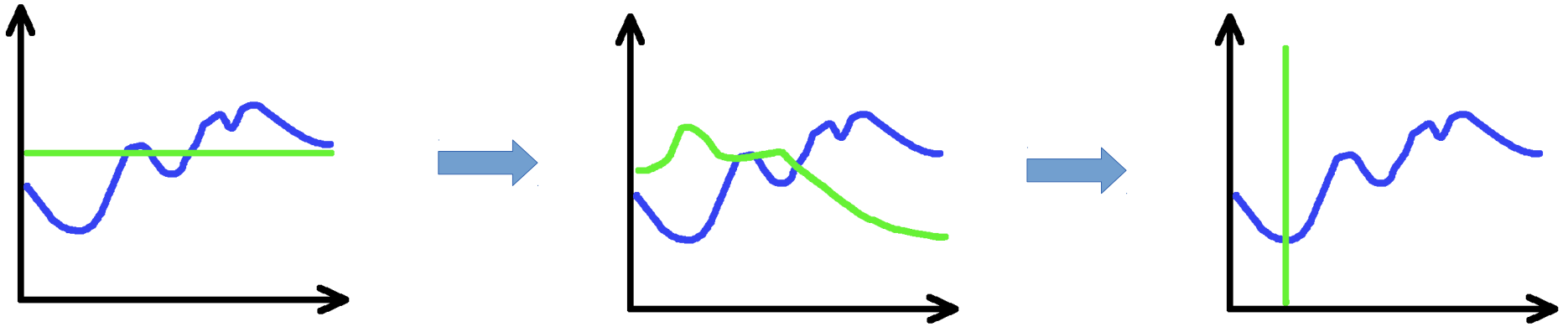
- Suppose H(0) is graph Laplacian.
  - Applications of H(0) hop to neighboring sites.
  - Ground state is uniform.



$$H(0) = nI - \sum_i \sigma_x^{(i)}$$

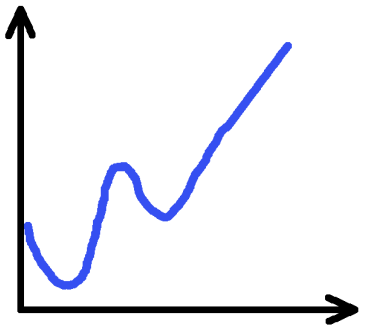
Adiabatic vs. Thermal

Thermal	Adiabatic
Probability distribution	Quantum Superposition
Decrease Temperature	Decrease Hopping Term
Quasi-static: stay in Gibbs distribution	Quasi-static: Stay in ground state



Adiabatic vs. Thermal

- Consider a one-dimensional double well:



- Thermal annealing: Boltzman factor
- Adiabatic optimization: tunneling matrix element
- These are different, so adiabatic can outperform thermal annealing.
- Better examples (on hypercube):
  [Farhi, Goldstone, Gutmann, 2002] [Reichardt, 2004]

Adiabatic Optimization

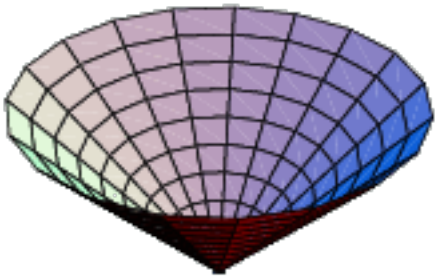
- Evaluating the eigenvalue gap is hard.
  - Numerics break down at n ~ 20-100
  - Analytic techniques mainly for high symmetry
  - Gap can be exponentially small even for easy problems with no local minima [Jarret, Jordan, 2014]
- What to do?
  - More math [cf. Altschuler, Krovi, Roland, 2007]
  - Try more general setting: go faster adiabatic theorem recommends and/or allow interaction with environment

    [cf. Nagaj, Somma, Kieferova, 2012]
  - Try experiments [cf. Boixo et al, 2013]

Hogg's Algorithm

Problem 4: Given black box for $f : \{0,1\}^n \to \mathbb{Z}$ with promise $f(x) = |x - y|$, find y using as few queries as possible.

- Classical: n queries
- Quantum: 1 query

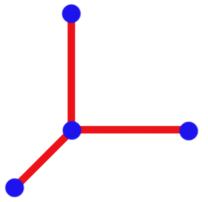


[Hogg, 1998]

Gradients and Quadratic Basins

Problem 5: Given black box for $f : \mathbb{R}^n \to \mathbb{R}$ find $\nabla f(0)$ using as few queries as possible.

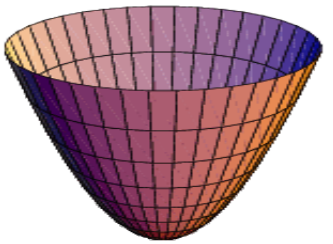
- Classical: n+1 queries
- Quantum: 1 query



[Jordan, 2005]

Problem 6: Given black box for $f : \mathbb{R}^n \to \mathbb{R}$ and promise $f(x) = x^t M x + b \cdot x + c$ with M positive definite, find minimum.

- Classical: $\Theta(n^2)$ queries
- Quantum: $O(n)$ queries



[Bulger, 2005]
[Jordan, 2008]

Very Recent Breakthrough

Problem 6 ("E3LIN2"): Given a list of N linear equations mod 2, satisfy as many as possible. Each equation involves exactly 3 variables. Each variable is in at most D equations.

- Classical: satisfy $\left(\frac{1}{2} + \frac{\text{constant}}{D}\right) N$
- Quantum: satisfy $\left(\frac{1}{2} + \frac{1}{22D^{3/4}}\right) N$
- NP-hard: satisfy $\left(\frac{1}{2} + \frac{\text{constant}}{D^{1/2}}\right) N$

[Farhi, Goldstone, Gutmann, 2014]

Summary

- Linear Algebra:
  - Can solve N linear equations in log N time
  - Needs special structure (e.g. sparsity)
- Sums and Integrals:
  - Can obtain precision ϵ in $1/\epsilon$ time.
  - Quadratic speedup by generalizing Grover's algorithm.
- Optimization:
  - Adiabatic optimization: promising but mysterious.
  - Factor of d speedups for symmetric d-dimensional optimization problems.
  - Recent breakthrough achieves better approximation factor for E3LIN2 than any classical polynomial-time algorithm.