Can a quantum computer solve optimization problems more efficiently than a classical computer?

Peter Young

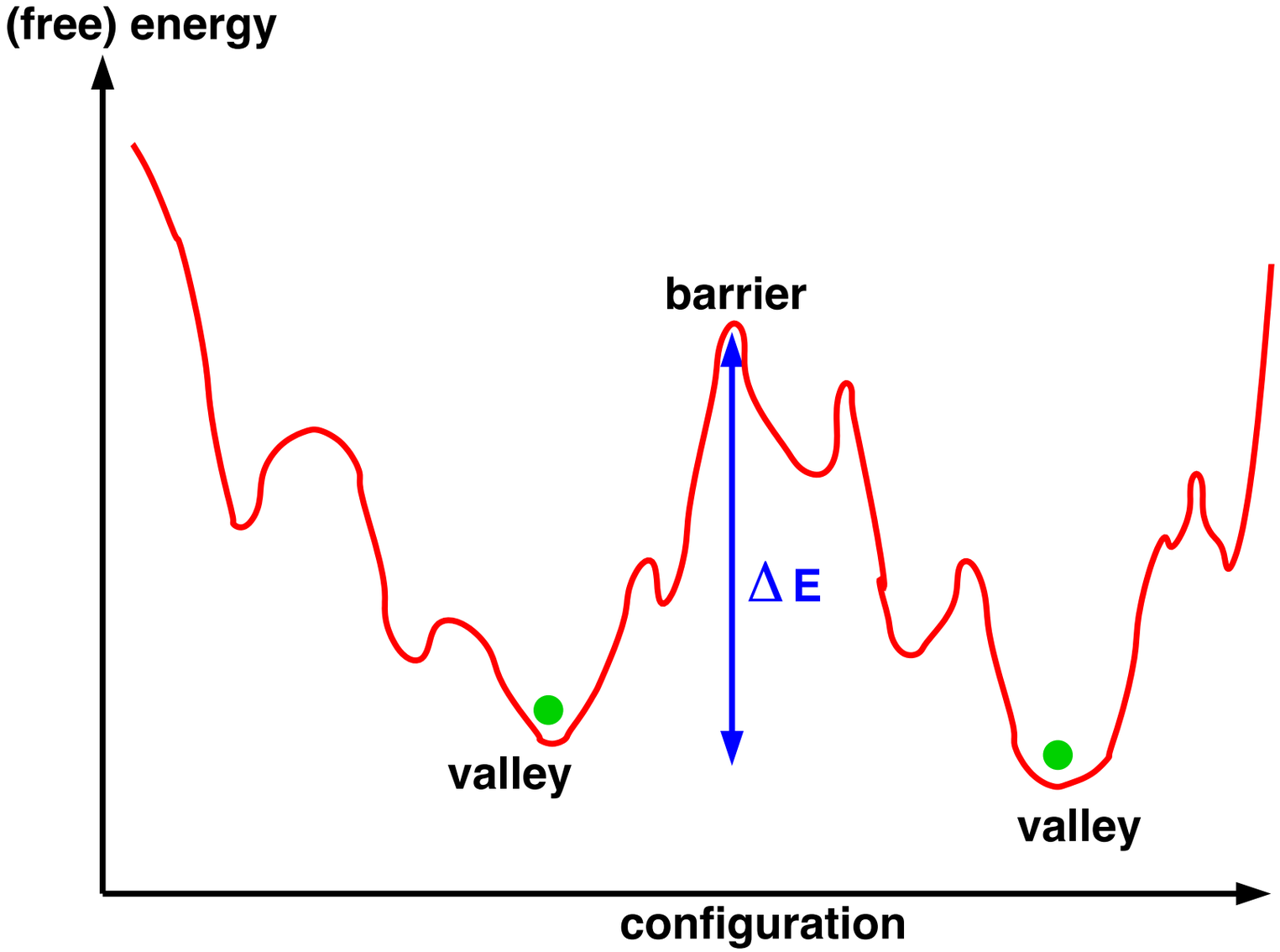
Colloquium at the Carl von Ossietzky University, Oldenburg, October 24, 2016

Overview

- What are optimization problems?
- Example, a spin glass.
- Classical, physics-inspired algorithm: simulated (thermal) annealing (SA).
- Introduction to quantum computing:
  - Gate model. Uses "quantum parallelism". Best example, the Shor algorithm for factoring integers. Must completely eliminate decoherence.
  - Quantum Annealing (QA). Uses "quantum tunneling". Hope is is somewhat insensitive to decoherence. **(Focus of this talk)**.
- Experiments on D-Wave machine (~ 1000 qubits on a board)
- Results of computer simulations to see if D-Wave gives a **quantum speedup**.
- Conclusions.

Optimization Problems

Minimize (or maximize) a function of many variables. We will call this "cost function" the energy. There is competition (which we will call "frustration") between different terms in the energy, so no configuration of the variables satisfies all the terms.



There is a complicated "energy landscape", so a simple (greedy) algorithm goes straight downhill in energy to a local minimum and is then stuck.

Examples of Optimization Problems

- Speech recognition (industry)
- Image recognition (industry)
- Finding the equilibrium (folded) configuration of proteins (biology)
- Solving “satisfiability” problems (computer science)
- Finding the ground state of a “spin glass” (see next slide) (physics)
- .....

Spin Glasses

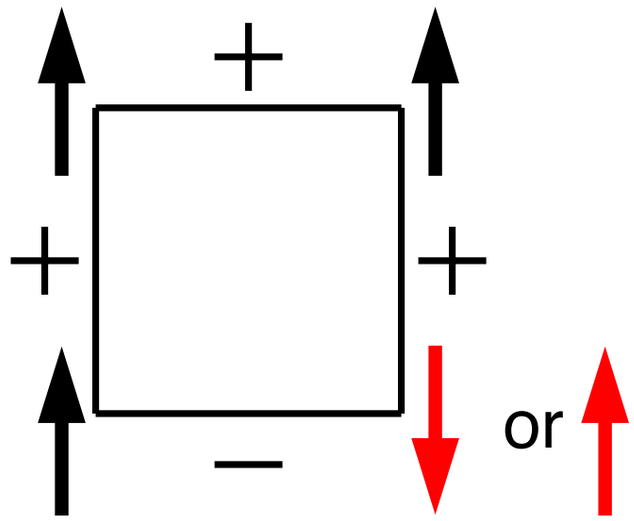
Spin glasses have been studied by physicists for many years. They are magnetic systems with disorder and “frustration”. They are convenient systems with which to study optimization algorithms because there a simple-to-write-down models which are amenable to computer simulation and can be implemented on quantum hardware (as we will see).

The standard model Hamiltonian (Edwards-Anderson, 1975) is

$$\mathcal{H} = -\sum_{\langle i,j \rangle} J_{ij} S_i S_j - \sum_i h_i S_i$$

where the S_i are Ising spins, ± 1, on a lattice, the J_{ij} are the “frustrated” interactions (random in sign). We may also include random longitudinal fields h_i.

Toy example on right. Bottom right spin can’t decide whether to be up or down.



Simulated (thermal) annealing (SA)

A physics inspired algorithm

Put in a temperature (Kirkpatrick et al, 1983) and simulate with Monte Carlo. Some probability of going up in energy to escape a local minimum.

Gradually reduce the temperature, so **$T(t) \rightarrow 0$ as $t \rightarrow \infty$**. If T decreases sufficiently slowly will reach the ground state.

Useful general-purpose algorithm. Here will use SA as a comparison with an analogous quantum algorithm, quantum annealing (QA).

Complexity

How much computer time is needed to solve the problem as a function of the size of the problem N?

- There are some problems which look complicated but for which there is a clever algorithm which solves the problem in a time proportional to a power of N, i.e. polynomial time.
  e.g. spin glass in two dimensions in zero field (in which the interactions form planar graph) (c.f. Hartmann).This is complexity class **P.**

- There is another set of problems, called complexity class **NP hard**, for which the time is exponential in N for all known algorithms, at least for large N and for the hardest instances at each size.
  e.g. spin glass in three or higher dimensions, and also two dimensions in a field or on a non-planar graph.
  No proof that a polynomial-time algorithm doesn't exist (unlikely)

Can quantum mechanics help?

A digression on quantum computing.

How is quantum different from classical? For our purposes in two ways: quantum parallelism and quantum tunneling. Each of these has given rise to a different paradigm for quantum computing. Will discuss each in turn.

Quantum Parallelism

A quantum state is a (coherent) linear superposition of basis states:

$$|\psi\rangle = \sum_{k=1}^{M} a_k |k\rangle$$

For systems with N 2-state *qubits*, M = 2^N. Acting on $|\psi\rangle$ with a unitary transformation (a gate) acts in parallel on all 2^N states. Can we gain from this parallelism? Problem is, to get information out we need to do a measurement: gives one result not 2^N.

Quantum Parallelism

Quantum Parallelism

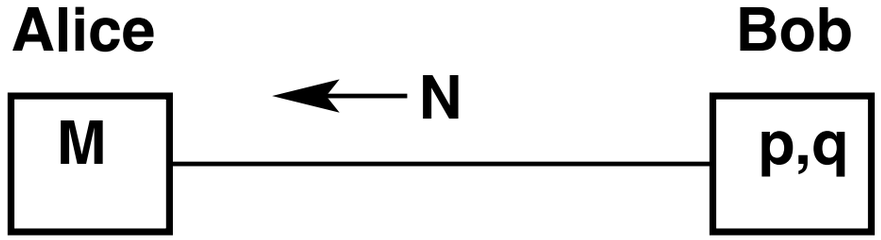
However, for some problems, by clever pre-processing **before** the measurement, answer can be found with a few runs of the algorithm. Most famous example is Shor's algorithm for factoring integers, i.e. N = p q (with p and q prime). Given N what are p and q? Potentially important because the difficulty of factoring is at the heart of a common method (RSA) of sending encrypted information down the internet. Here's a simplified version of RSA:

Alice wants to send a message M to Bob down a public channel but it must be encrypted to M' so only Bob can read it.

- Bob sends to Alice N (the public key) but keeps p and q (private key) to himself.
- Alice uses N to encode message, i.e. M → M', and sends M' to Bob.
- Bob uses his private key, p and q separately, to decode the message, i.e. M' → M. (The message can not be decoded knowing only the public key).

Quantum Parallelism

However, for some problems, by clever pre-processing **before** the measurement, answer can be found with a few runs of the algorithm. Most famous example is Shor's algorithm for factoring integers, i.e. N = p q (with p and q prime). Given N what are p and q?
Potentially important because the difficulty of factoring is at the heart of a common method (RSA) of sending encrypted information down the internet. Here's a simplified version of RSA:



Alice wants to send a message M to Bob down a public channel but it must be encrypted to M' so only Bob can read it.

- Bob sends to Alice N (the public key) but keeps p and q (private key) to himself.
- Alice uses N to encode message, i.e. M → M', and sends M' to Bob.
- Bob uses his private key, p and q separately, to decode the message, i.e. M' → M. (The message can not be decoded knowing only the public key).

Quantum Parallelism

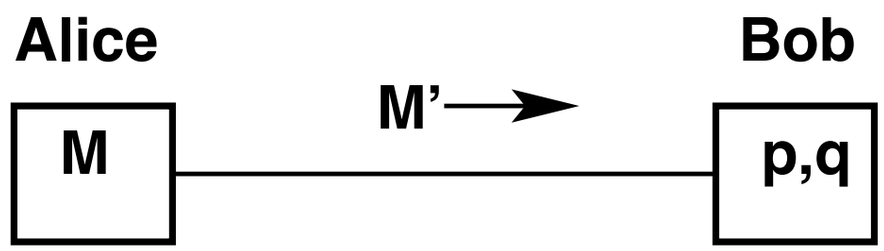
However, for some problems, by clever pre-processing **before** the measurement, answer can be found with a few runs of the algorithm. Most famous example is Shor's algorithm for factoring integers, i.e. N = p q (with p and q prime). Given N what are p and q? Potentially important because the difficulty of factoring is at the heart of a common method (RSA) of sending encrypted information down the internet. Here's a simplified version of RSA:

Alice wants to send a message M to Bob down a public channel but it must be encrypted to M' so only Bob can read it.

- Bob sends to Alice N (the public key) but keeps p and q (private key) to himself.
- Alice uses N to encode message, i.e. M → M', and sends M' to Bob.
- Bob uses his private key, p and q separately, to decode the message, i.e. M' → M. (The message can not be decoded knowing only the public key).

Quantum Parallelism

However, for some problems, by clever pre-processing **before** the measurement, answer can be found with a few runs of the algorithm. Most famous example is Shor's algorithm for factoring integers, i.e. N = p q (with p and q prime). Given N what are p and q?
Potentially important because the difficulty of factoring is at the heart of a common method (RSA) of sending encrypted information down the internet. Here's a simplified version of RSA:



Alice wants to send a message M to Bob down a public channel but it must be encrypted to M' so only Bob can read it.

- Bob sends to Alice N (the public key) but keeps p and q (private key) to himself.
- Alice uses N to encode message, i.e. M → M', and sends M' to Bob.
- Bob uses his private key, p and q separately, to decode the message, i.e. M' → M. (The message can not be decoded knowing only the public key).

Quantum Parallelism

However, for some problems, by clever pre-processing **before** the measurement, answer can be found with a few runs of the algorithm. Most famous example is Shor's algorithm for factoring integers, i.e. N = p q (with p and q prime). Given N what are p and q? Potentially important because the difficulty of factoring is at the heart of a common method (RSA) of sending encrypted information down the internet. Here's a simplified version of RSA:

Alice wants to send a message M to Bob down a public channel but it must be encrypted to M' so only Bob can read it.

- Bob sends to Alice N (the public key) but keeps p and q (private key) to himself.
- Alice uses N to encode message, i.e. M → M', and sends M' to Bob.
- Bob uses his private key, p and q separately, to decode the message, i.e. M' → M. (The message can not be decoded knowing only the public key).

Quantum Parallelism

However, for some problems, by clever pre-processing **before** the measurement, answer can be found with a few runs of the algorithm. Most famous example is Shor's algorithm for factoring integers, i.e. N = p q (with p and q prime). Given N what are p and q?
Potentially important because the difficulty of factoring is at the heart of a common method (RSA) of sending encrypted information down the internet. Here's a simplified version of RSA:

Alice wants to send a message M to Bob down a public channel but it must be encrypted to M' so only Bob can read it.

- Bob sends to Alice N (the public key) but keeps p and q (private key) to himself.
- Alice uses N to encode message, i.e. M → M', and sends M' to Bob.
- Bob uses his private key, p and q separately, to decode the message, i.e. M' → M. (The message can not be decoded knowing only the public key).

Shor's Algorithm

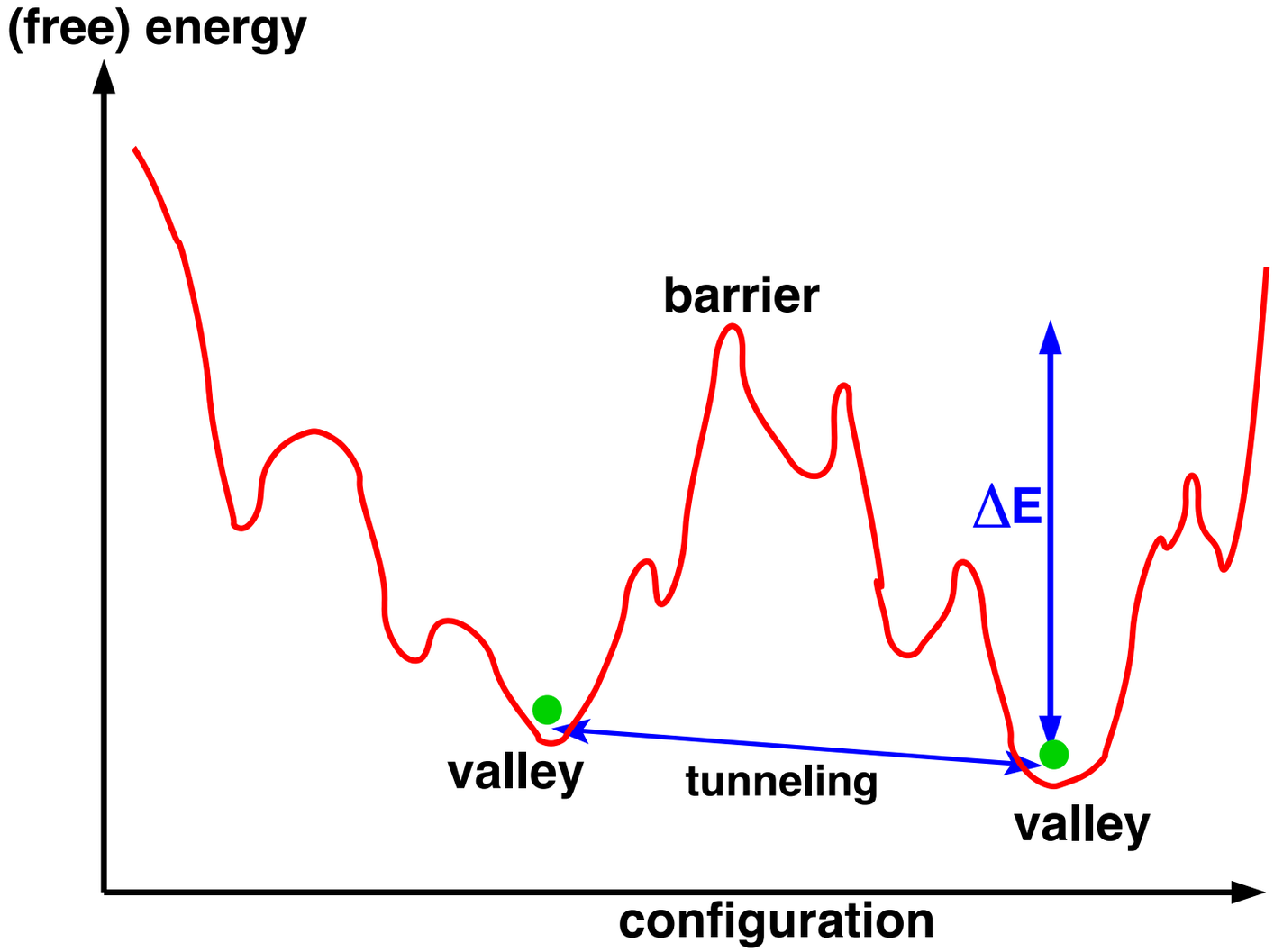
There is a lot of number theory behind Shor's algorithm. The quantum part only comes in finding the period of a certain function. This is done by a quantum Fourier transform.
To factor an n-bit integer Shor's algorithm requires $O(n^3)$ operations. The best-known classical algorithm takes of order $\exp(\text{const. } n^{1/3})$ operations. The polynomial quantum algorithm wins heavily for large n.

Problem: depends crucially on coherence. Even a small amount of decoherence kills the quantum parallelism. Experimentally, there is always noise, i.e. decoherence. Shor's algorithm has only been implemented for a very small number of qubits (~5) and 15 was successfully factored.
There are error correcting codes (Shor again, and others) but, still needs intrinsic error rate to be low, and requires additional qubits.

BUT: if the experimental problems could be overcome we **know** that there **is** a **quantum speedup**.

Quantum Tunneling

The second aspect in which quantum is different from classical, and which gives rise to a second paradigm of quantum computing, is quantum tunneling. (The focus of the rest of the talk)



Rather than being thermally activated over a barrier a quantum particle can tunnel through it.

Hence try quantum annealing (QA), like simulated annealing (SA) but using quantum rather than thermal, fluctuations to overcome barriers.

Quantum Annealing (QA)

Hope that we still get tunneling, even multi-particle tunneling, even if there is some decoherence, i.e. less sensitive to decoherence than Shor's algorithm. As we shall see there are experiments with ~ 1000 qubits, which certainly do not maintain quantum coherence during the evolution of the algorithm but seem to have quantum behavior (at least to some extent).

BUT: unlike Shor's algorithm we have no guarantee of a quantum speedup even on a perfect quantum annealer. Tunneling is likely to be better than thermal activation when barriers are high but thing (think of the WKB formula). But are real problems like this?

Make the model quantum

Before we had Ising spins S_i which take values ± 1. Now make them quantum operators $\boldsymbol{\sigma_i^z}$ (Pauli spin matrices) and work in the basis in which these are diagonal (the computational basis) and so they also have values ± 1. So far the model is unchanged.
The simplest way to induce quantum fluctuations is to add a transverse field h^T involving the $\boldsymbol{\sigma_i^x}$. Our spin glass Hamiltonian is therefore

$$\mathcal{H} = -\sum_{\langle i,j \rangle} J_{ij}\sigma_i^z\sigma_j^z - \sum_i h_i\sigma_i^z - h^T\sum_i \sigma_i^x$$

Since $\boldsymbol{\sigma_i^x}$ and $\boldsymbol{\sigma_i^z}$ don't commute we have quantum fluctuations. h^T is like temperature, make it large initially and then slowly decrease it with time so we end up in the ground state of the spin glass Hamiltonian (the first two terms, those involving the $\boldsymbol{\sigma_i^z}$).

Quantum Adiabatic Algorithm (QAA)

A variation on quantum annealing, also inspired by physics (Farhi et al, 2001). Imagine running QA annealing on a real device with programmable couplings (such a device exists as will see in next slide). Start with only the transverse field term, and prepare the qubits in the ground state, spins along x. Then, in real time, slowly decrease the transverse field piece and increase (from zero) the spin glass part until, at the end, there is no transverse field term (only the spin glass). The adiabatic theorem of quantum mechanics tells us that if the evolution is slow enough the system stays in its instantaneous ground state , and so we end up in the ground state of the spin glass. The problem is solved!

But, how slowly do we have to go as a function of the problem size N?

A company, D-Wave has produced a machine to implement the QAA. The latest version has ~1000 qubits (next slide).

D-Wave

D-Wave: large number of superconducting qubits on a board at milliKelvin temp.

Latest version ~1000 qubits

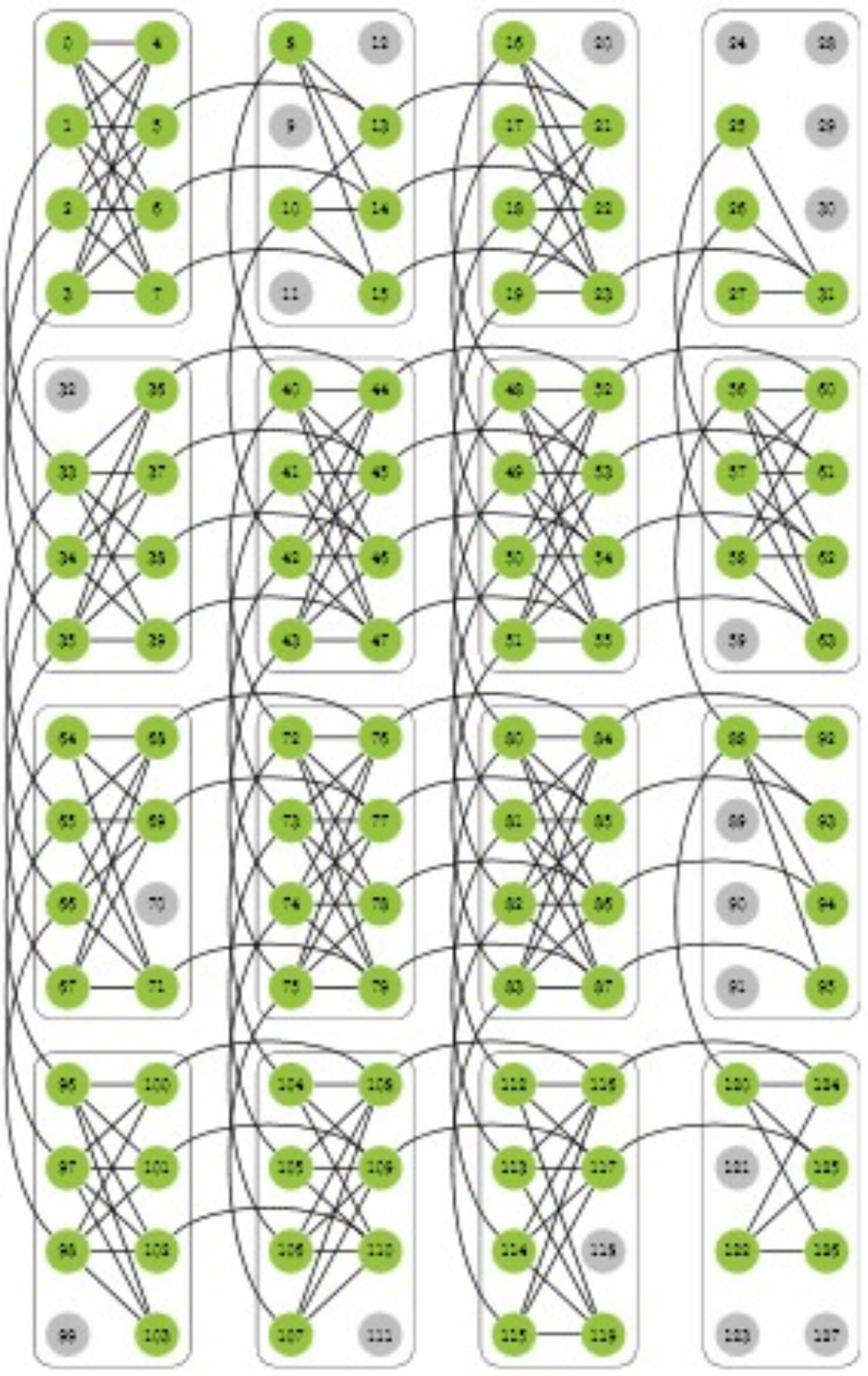
Runs the QAA.

During the run, phase coherence is not maintained, hence call this a quantum annealer.

Questions:

- D-Wave has noise and non-zero T, so is it really quantum?
- If it is, then is the D-Wave machine more efficient than a classical computer?

D-Wave



Connections of the qubits form a (2-d) “chimera” graph, see figure for D-Wave 1 (128 qubits, not all functional).

Comparison between D-Wave and SA

(Rønnow et al. arXiv:1401.2910, Science **345**, 420 (2014).)

Consider a spin glass on the chimera graph (so the problem fits naturally on to the D-Wave machine, this version ~500 qubits).

Do efficient simulated annealing (SA) on a computer, and compare with runs on D-Wave, for different sizes N.

Is there a quantum speedup?

Not so trivial to determine because:

(i) Need to determine optimal annealing schedule

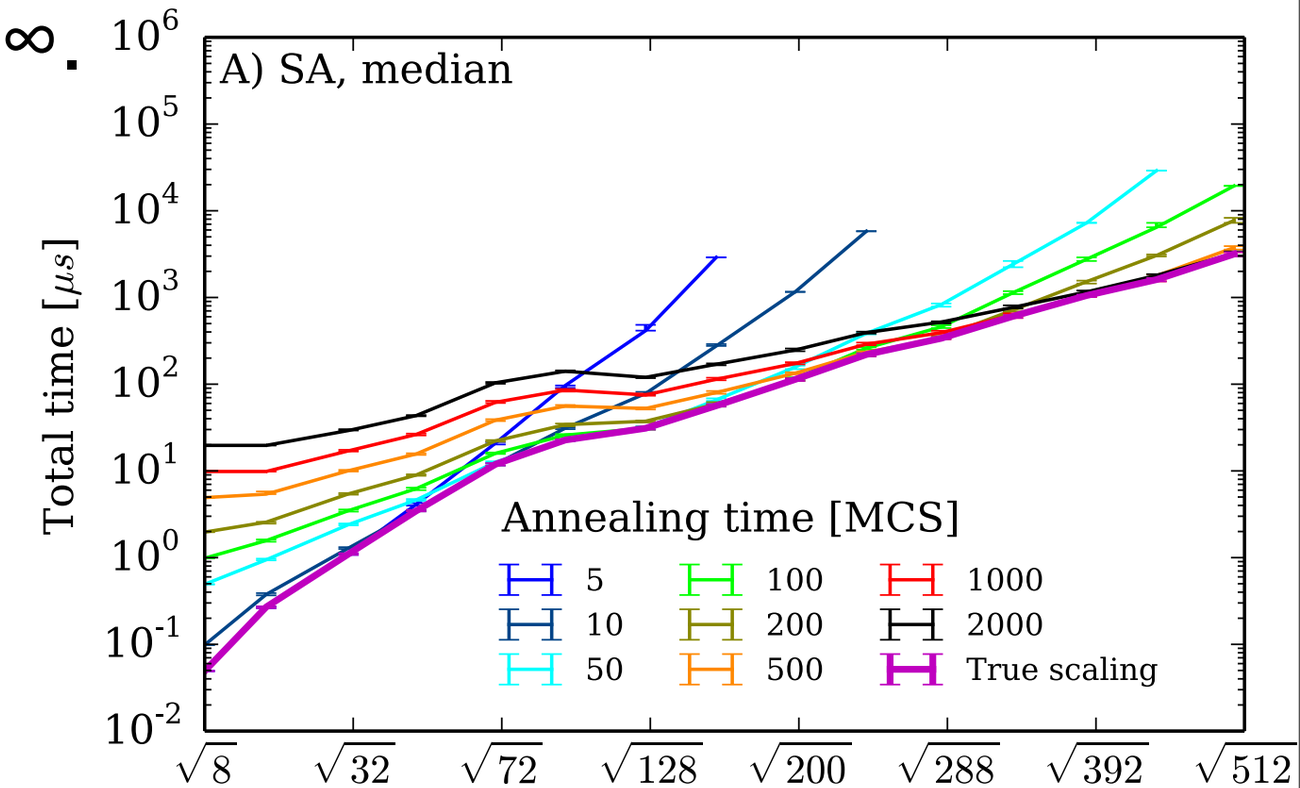
(ii) Runtime depends on the specific instance

(iii) Need to extrapolate to N = ∞.

Consider (i)

Figure shows SA.

For D-Wave, minimum annealing time (20μs) is longer than optimal.



Comparison (continued)

Now consider (ii) and (iii) (instance dependence and extrapolation to N = ∞).

(Rønnow et al.)

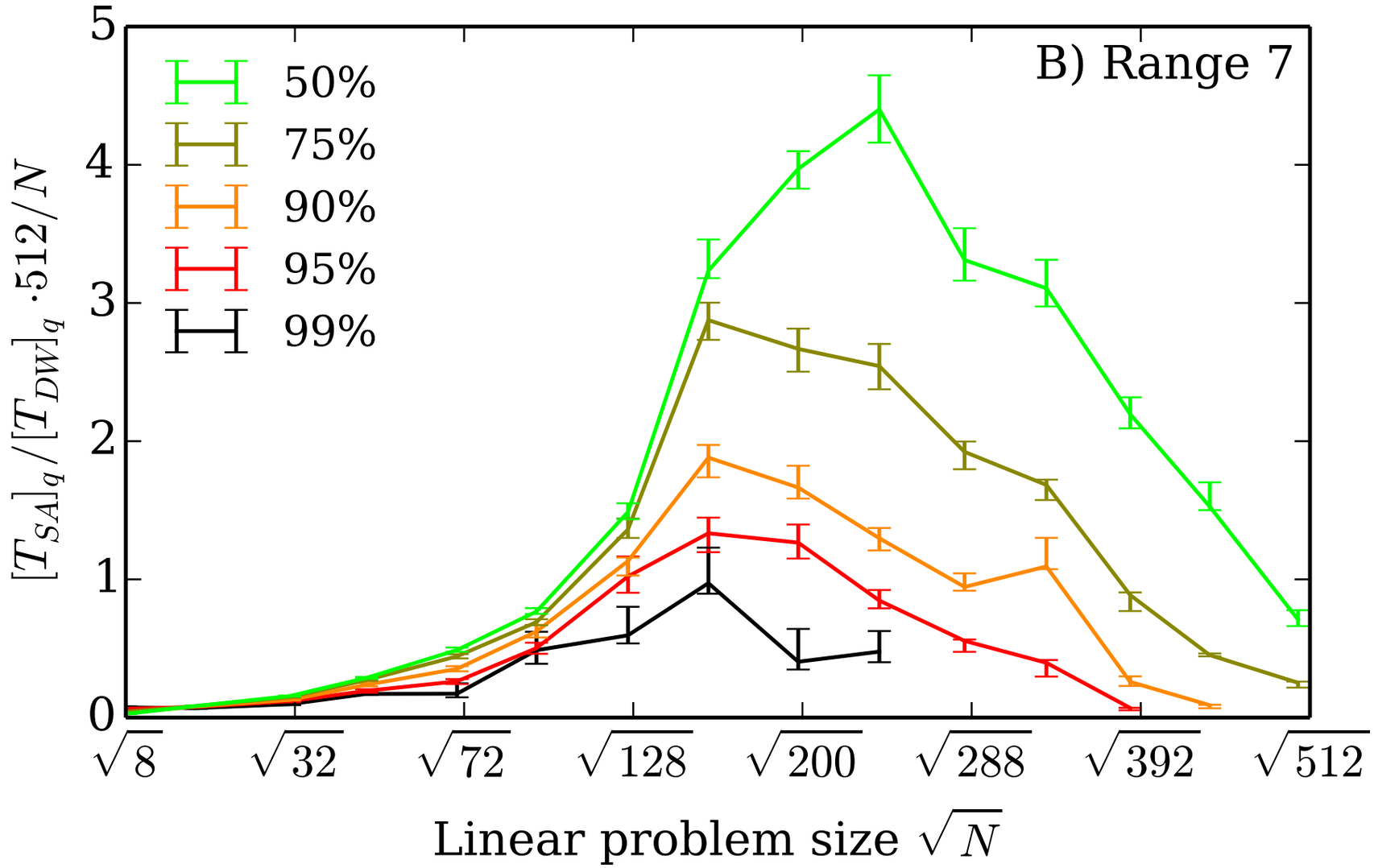


Figure is ratio T_{SA} / T_{DW}. If increases at large N, evidence for a quantum speedup. The "%" is a percentile, indicating fraction of instances that have been solved, so "50%" (green) is the median. For black points, all but 1% of the instances have been solved. Data does not show evidence for a quantum speedup.

Some more relevant physics (chaos)

As T is lowered in SA the spin glass configuration that that minimizes the free energy can change (quite suddenly, a rounded "transition") which is called temperature chaos, or T-chaos for short. Spin correlations change at distances greater than ℓ where

$$\ell = c_T \, (\Delta T)^{-\zeta}$$

Similarly, in QA there is chaos with respect to h^T.

In addition to T-chaos (in SA) and TF-chaos (in QA), there is also sensitivity to small changes in the interactions, called J-chaos, where the length scale is $\ell = c_J (\Delta J)^{-\zeta}$

Numerically $\zeta \simeq 1$ in d = 2, 3, 4 for both J-chaos and T-chaos.

However, the amplitude is much bigger for J-chaos, i.e.

$$c_J \gg c_T$$

Example of T-chaos on the chimera graph

In some spin glass samples T-chaos will not occur, in others it may occur once, twice etc. Instances where this occurs will be particularly hard to solve. Fraction of instances where T-chaos occurs is found to increase with increasing size N.

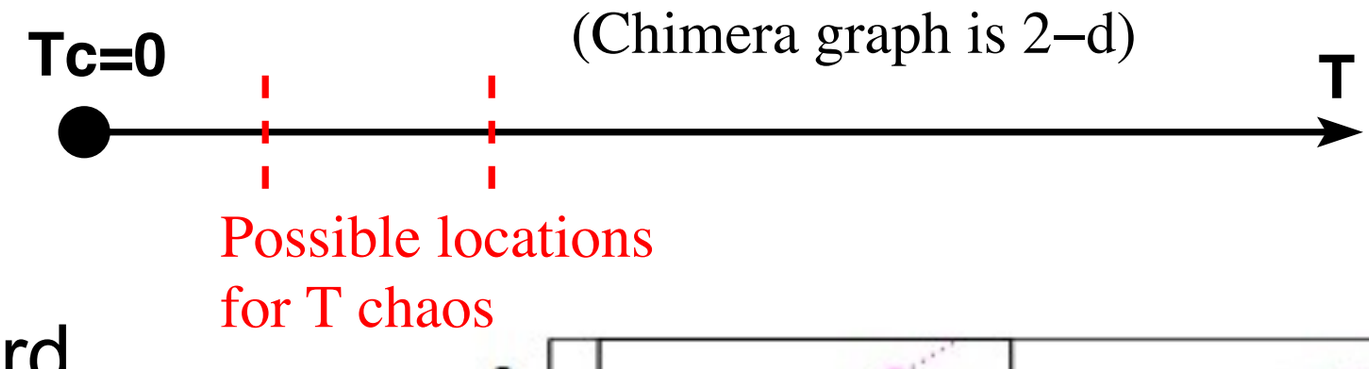
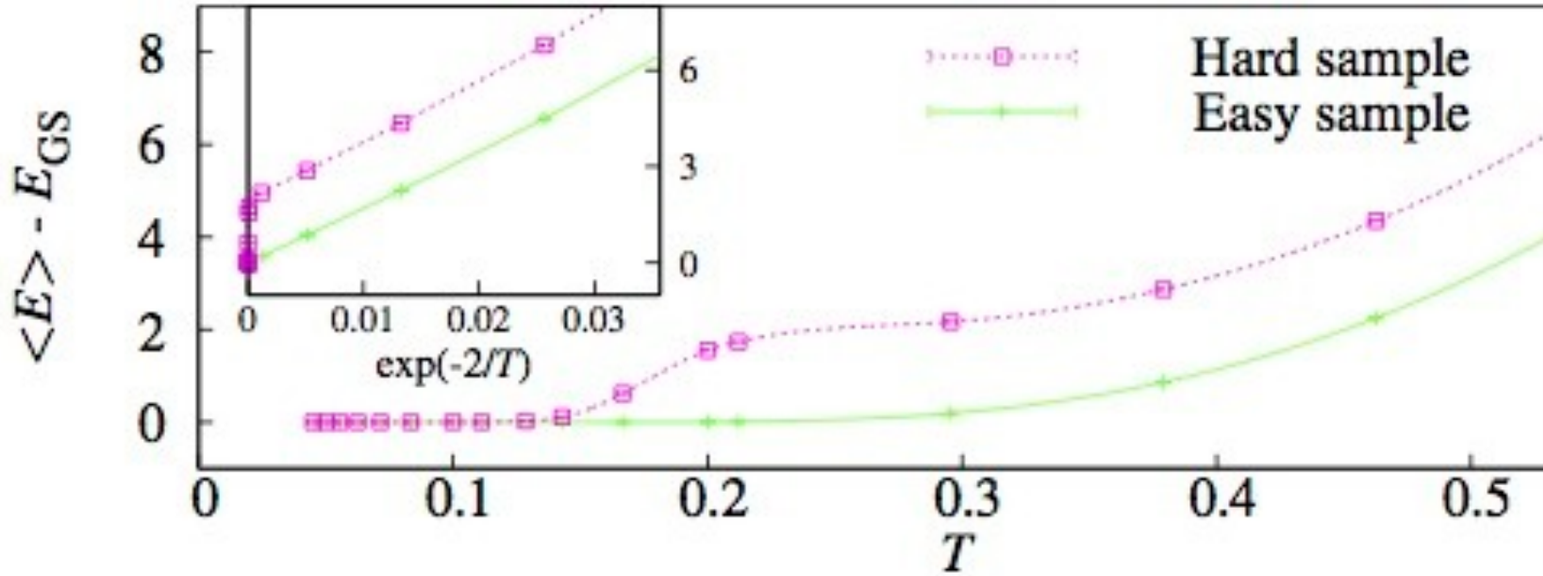


Figure shows a hard sample, in which the energy shows a pronounced change at low-T due to temperature chaos, and an easy sample where this does not occur.



(From Martin-Mayor and Hen, arXiv:1502.02494)

Sample-to-sample fluctuations

There is a **broad distribution** in the values of the time to solution τ. Interpretation: samples with small τ presumably have no T-chaos, for SA ,or TF-chaos for QA, while those with large τ presumably have one or more temperatures where T-chaos occurs.

One finds that T-chaos is rare for small sizes but happens in most samples for very large sizes.

T-chaos is problematic for classical, annealing-type algorithms.

- Is TF-chaos a problem quantum annealers?
- Are instances with T-chaos (in SA) also those with TF-chaos (in QA)?

Needs more work to see.

Limitations of the D-Wave machine

- The temperature may not be low enough. For instances where temperature chaos occurs at a temperature lower than that of the chip then the wrong answer will typically be obtained.
- The strengths of the bonds are not represented exactly in the (analog) D-Wave machine (intrinsic control errors, ICE). Even small changes in the bond strengths can dramatically change the ground state. This is called "J-chaos". Thus D-Wave machine might be getting the right ground state to the wrong problem (some of the time). Do samples with strong T-chaos also have strong J-chaos? Probably, but more work needed to make this precise.
- Non-thermal noise in the superconducting qubits. Needs to be understood better.

Conclusions

Conclusions

- There are two paradigms for quantum computing

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.
  - Quantum annealing (focus of this talk). Less susceptible to decoherence (we think, but need to understand better) and there is a device with ~1000 qubits, but no provable quantum speedup.

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.
  - Quantum annealing (focus of this talk). Less susceptible to decoherence (we think, but need to understand better) and there is a device with ~1000 qubits, but no provable quantum speedup.
- "Chaos" plays a role for both QA and SA.

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.
  - Quantum annealing (focus of this talk). Less susceptible to decoherence (we think, but need to understand better) and there is a device with ~1000 qubits, but no provable quantum speedup.
- "Chaos" plays a role for both QA and SA.
- Is there, in practice, a quantum speedup in QA (QAA)?:

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.
  - Quantum annealing (focus of this talk). Less susceptible to decoherence (we think, but need to understand better) and there is a device with ~1000 qubits, but no provable quantum speedup.
- "Chaos" plays a role for both QA and SA.
- Is there, in practice, a quantum speedup in QA (QAA)?:
  - Not trivial to determine because need to optimize the annealing time (hasn't yet been possible with D-Wave machine) and need to extrapolate to large N and average (in some way) over instances.

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.
  - Quantum annealing (focus of this talk). Less susceptible to decoherence (we think, but need to understand better) and there is a device with ~1000 qubits, but no provable quantum speedup.
- "Chaos" plays a role for both QA and SA.
- Is there, in practice, a quantum speedup in QA (QAA)?:
  - Not trivial to determine because need to optimize the annealing time (hasn't yet been possible with D-Wave machine) and need to extrapolate to large N and average (in some way) over instances.
  - So far, though, no evidence for quantum speedup on the usually studied problems (some problems have been cooked up to be especially hard for simple implementations of SA and a resulting speedup has been claimed).

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.
  - Quantum annealing (focus of this talk). Less susceptible to decoherence (we think, but need to understand better) and there is a device with ~1000 qubits, but no provable quantum speedup.
- "Chaos" plays a role for both QA and SA.
- Is there, in practice, a quantum speedup in QA (QAA)?:
  - Not trivial to determine because need to optimize the annealing time (hasn't yet been possible with D-Wave machine) and need to extrapolate to large N and average (in some way) over instances.
  - So far, though, no evidence for quantum speedup on the usually studied problems (some problems have been cooked up to be especially hard for simple implementations of SA and a resulting speedup has been claimed).

Danke schön

Conclusions

Conclusions

- There are two paradigms for quantum computing

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.
  - Quantum annealing (focus of this talk). Less susceptible to decoherence (we think, need to understand better) and there is a device with ~1000 qubits, but no provable quantum speedup.

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.
  - Quantum annealing (focus of this talk). Less susceptible to decoherence (we think, need to understand better) and there is a device with ~1000 qubits, but no provable quantum speedup.
- Is there, in practice, a quantum speedup in QA (QAA)?:

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.
  - Quantum annealing (focus of this talk). Less susceptible to decoherence (we think, need to understand better) and there is a device with ~1000 qubits, but no provable quantum speedup.
- Is there, in practice, a quantum speedup in QA (QAA)?:
  - Not trivial to determine because need to optimize the annealing time (hasn't been possible to do with D-Wave machine) and need to extrapolate to large N.

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.
  - Quantum annealing (focus of this talk). Less susceptible to decoherence (we think, need to understand better) and there is a device with ~1000 qubits, but no provable quantum speedup.
- Is there, in practice, a quantum speedup in QA (QAA)?:
  - Not trivial to determine because need to optimize the annealing time (hasn't been possible to do with D-Wave machine) and need to extrapolate to large N.
  - So far, though, no evidence for quantum speedup on the usually studied problems (some problems have been cooked up to be especially hard to simple implementations of SA and a resulting speedup has been claimed).

Conclusions

- There are two paradigms for quantum computing
  - Gate model. There are some algorithms (e.g. Shor) where there is a provable quantum speedup, but they are very susceptible to decoherence so application has been limited to a small number of qubits so far.
  - Quantum annealing (focus of this talk). Less susceptible to decoherence (we think, need to understand better) and there is a device with ~1000 qubits, but no provable quantum speedup.
- Is there, in practice, a quantum speedup in QA (QAA)?:
  - Not trivial to determine because need to optimize the annealing time (hasn't been possible to do with D-Wave machine) and need to extrapolate to large N.
  - So far, though, no evidence for quantum speedup on the usually studied problems (some problems have been cooked up to be especially hard to simple implementations of SA and a resulting speedup has been claimed).

Danke schön