Complexity of the Quantum Adiabatic Algorithm

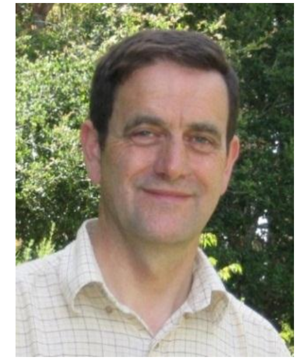
Itay Hen

Adiabatic Quantum Computing 2013
March 6-8, 2013

arXiv preprints: 1109.6872, 1112.2269
1207.1712, 1208.3757

Collaborators

- joint work with Peter Young, UC Santa Cruz



- also partly with
  Eddie Farhi, Peter Shor,
  David Gosset, M.I.T.



- Anders Sandvik, Boston University

- and Francesco Zamponi,
  Ecole Normale Supérieure.

Motivation

in what ways quantum computers are more efficient than classical computers?

what problems could be solved more efficiently on a quantum computer?

- best-known examples are:
  - Shor's algorithm for integer factorization. solves the problem in polynomial time (exponential speedup).
    current quantum computers can factor all integers up to *21*.
  - Grover's algorithm is a quantum algorithm for searching an unsorted database with N entries in $O\left(N^{1/2}\right)$ time (quadratic speedup).
- importance is huge (cryptanalysis, etc.).

Motivation

what more can quantum computers do?

- here, we will discuss “hard” satisfiability (SAT) and optimization problems which are at least “NP-complete”, i.e.,
  - hard to solve classically; time needed is exponential in the input (exponential complexity).
  - could a quantum computer solve these problems in an efficient manner? perhaps even in polynomial time?
- we also discuss the graph isomorphism problem.

the Quantum Adiabatic Algorithm is a general approach to solve a broad range of hard optimization problems using a quantum computer [Farhi et al.,2001]

Outline

- introduction:
  - the quantum adiabatic algorithm (QAA)
- satisfiability problems
  - the specific SAT models studied
  - method: quantum Monte Carlo simulations
  - results: complexity of the quantum adiabatic algorithm
  - results: comparison to the classical algorithm WalkSAT
- 3reg Max-Cut: an antiferromagnet on a random graph
- QAA, applied to the graph isomorphism problem
- summary, conclusions and future research

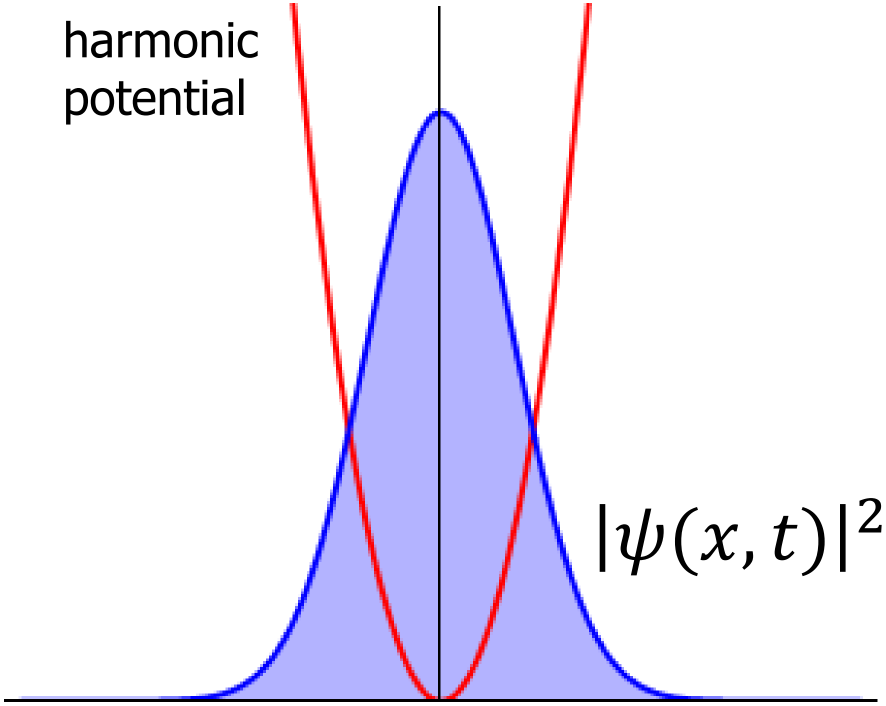
The Quantum Adiabatic Algorithm (QAA)

The adiabatic theorem of QM

- the adiabatic theorem of QM tells us that a physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.

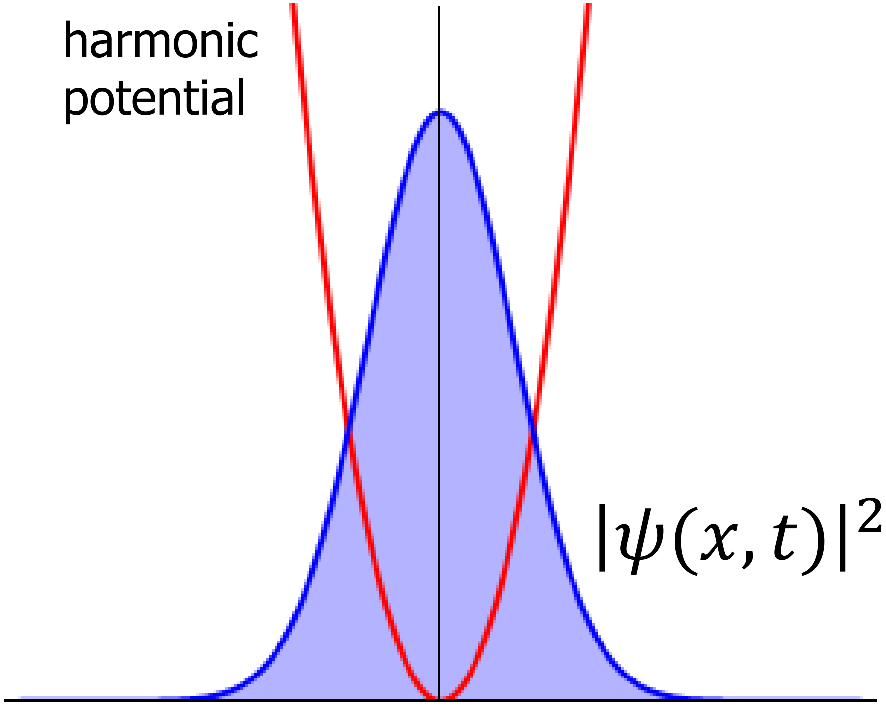
The adiabatic theorem of QM

- the adiabatic theorem of QM tells us that a physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.
- example: change the strength of a harmonic potential of a system in the ground state:



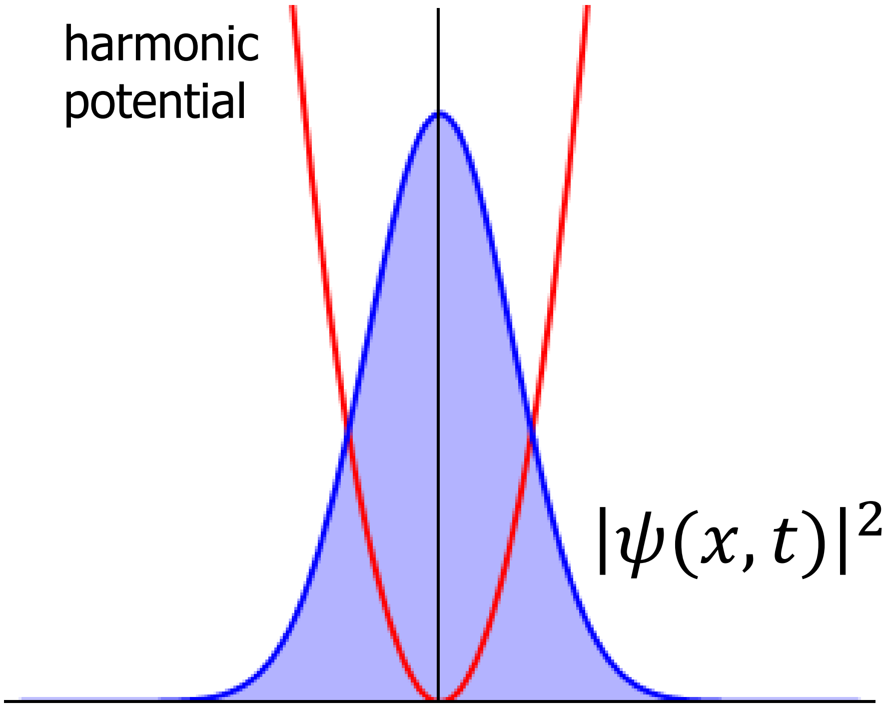
The adiabatic theorem of QM

- the adiabatic theorem of QM tells us that a physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.
- example: change the strength of a harmonic potential of a system in the ground state:
- an abrupt change (a *diabatic* process):



The adiabatic theorem of QM

- the adiabatic theorem of QM tells us that a physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.
- example: change the strength of a harmonic potential of a system in the ground state:
- a gradual slow change (an *adiabatic* process): wave function can "keep up" with the change.



The quantum adiabatic algorithm (QAA)

- the mechanism proposed by Farhi et al., the QAA:

1. take a difficult (classical) optimization problem.
2. encode its solution in the ground state of a quantum "problem" Hamiltonian $\hat{H}_p$.
3. prepare the system in the ground state of another easily solvable "driver" Hamiltonian" $\hat{H}_d$. $[\hat{H}_p, \hat{H}_d] \neq 0$.
4. vary the Hamiltonian slowly and smoothly from $\hat{H}_d$ to $\hat{H}_p$ until ground state of $\hat{H}_p$ is reached.

The quantum adiabatic algorithm (QAA)

- the interpolating Hamiltonian is this:

$$\hat{H}(t) = s(t)\hat{H}_p + [1 - s(t)]\hat{H}_d$$

$\hat{H}_p$ is the problem Hamiltonian whose ground state encodes the solution of the optimization problem

$\hat{H}_d$ is an easily solvable driver Hamiltonian, which does not commute with $\hat{H}_p$

- the parameter s obeys $0 \le s(t) \le 1$, with $s(0) = 0$ and $s(\mathcal{T}) = 1$. also: $\hat{H}(0) = \hat{H}_d$ and $\hat{H}(\mathcal{T}) = \hat{H}_p$.
- here, t stands for time and $\mathcal{T}$ is the running time, or complexity, of the algorithm.

The quantum adiabatic algorithm (QAA)

- the interpolating Hamiltonian is this:

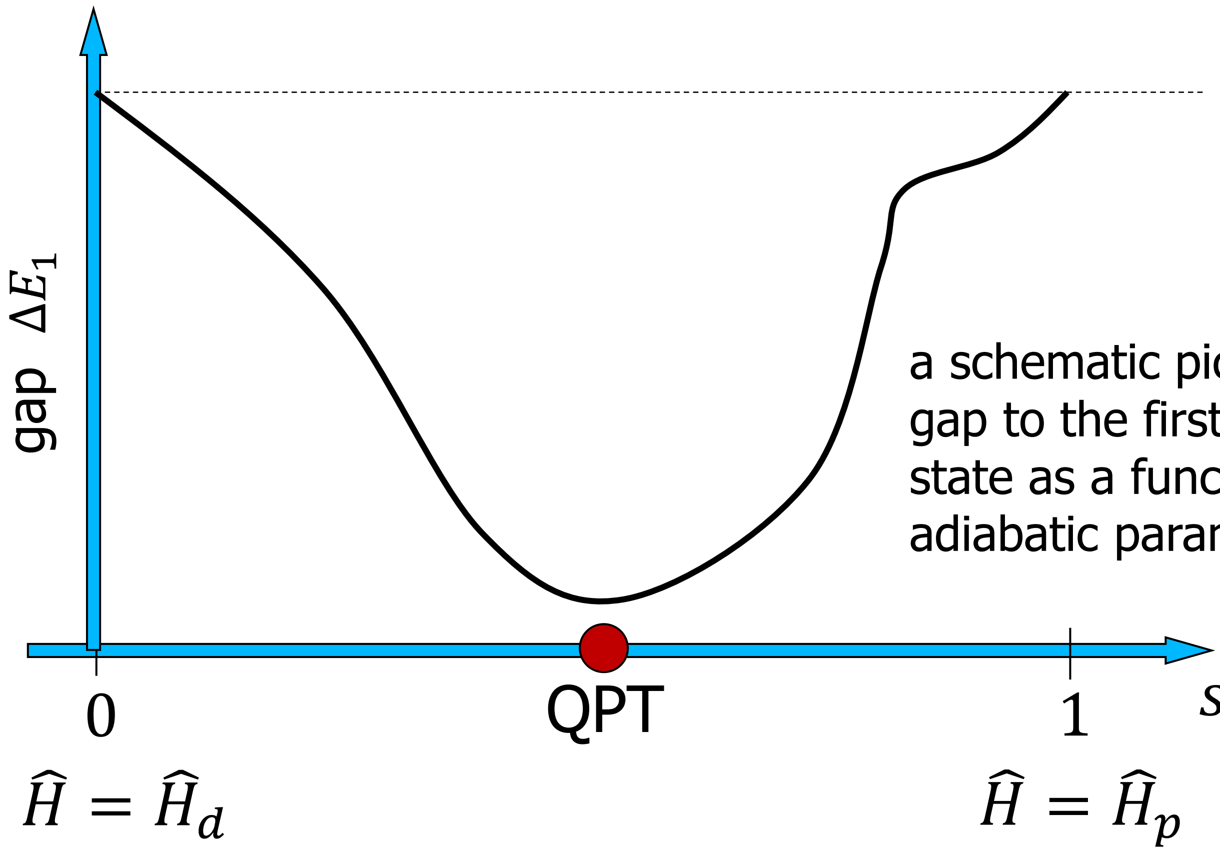
$$\hat{H}(t) = s(t)\hat{H}_p + [1 - s(t)]\hat{H}_d$$

- the adiabatic theorem ensures that if the change in $s(t)$ is made slowly enough, the system will stay close to the ground state of the instantaneous Hamiltonian throughout the evolution.
- one finally obtains a state close to the ground state of $\hat{H}_p$.
- measuring the state will give the solution of the original problem with high probability.

how fast can the process be?

Quantum phase transition

- bottleneck is likely to be a quantum phase transition (QPT) where the gap to the first excited state is small.
- there, the probability to "get off track" is maximal.



a schematic picture of the gap to the first excited state as a function of the adiabatic parameter s.

Quantum phase transition

- Landau-Zener theory tells us that to stay in the ground state the running time needed is:

$$\mathcal{T} \propto {}^{1}/_{\Delta E_{min}^{2}}$$

- exponentially closing gap (as a function of problem size N) → exponentially long running time → exponential complexity.

Quantum phase transition

- Landau-Zener theory tells us that to stay in the ground state the running time needed is:

$$\mathcal{T} \propto {}^{1}/_{\Delta E_{min}^{2}}$$

- exponentially closing gap (as a function of problem size N) → exponentially long running time → exponential complexity.

- it would be interesting to explore what one can do with "Local Adiabatic Evolution", i.e., by slowing down when approaching the minimum gap etc. for example, the adiabatic algorithm for Grover's search problem.

The quantum adiabatic algorithm

- most interesting unknown about QAA to date:

could the QAA solve in polynomial time "hard" (NP-complete) problems?

or

for which hard problems is $\mathcal{T}$ sub-exp' in N?

The quantum adiabatic algorithm

- most interesting unknown about QAA to date:

could the QAA solve in polynomial time "hard" (NP-complete) problems?

or

for which hard problems is $\mathcal{T}$ sub-exp' in N?

- early studies [Farhi et al., Hogg] on very small systems (number of bits $N \leq 24$) seemed to indicate that for some problems complexity scales like N^2.
- later studies on bigger systems showed a "crossover" from polynomial to exponential complexity [Young et al.].
- matter is still in debate [Altshuler et al., Knysh and Smelyanskiy]. no clear-cut example. Sergey's latest result.

The quantum adiabatic algorithm

- another interesting unknown:

what is the future of
adiabatic quantum computation?

- almost no examples that adiabatic quantum computation is efficient. however there exists an adiabatic version of Grover's search algorithm that uses "local adiabatic evolution".

- also, there is a correspondence between circuit-based computing and adiabatic computing [Aharonov et al., 2005].

- also, D-Wave Systems have built operational prototypical quantum annealers based on superconductor flux qubits (still being debated whether really quantum or not).

Satisfiability problems

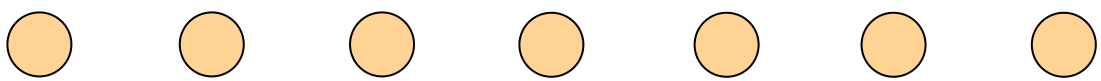
Satisfiability problems

- here, we consider certain "constraint satisfaction" (SAT) problems that are known to be hard classically.
- in SAT problems we ask whether there is an assignment of N bits (or Ising spins) which satisfies all of M clauses (or logical conditions). bits in each clause are chosen at random.

Satisfiability problems

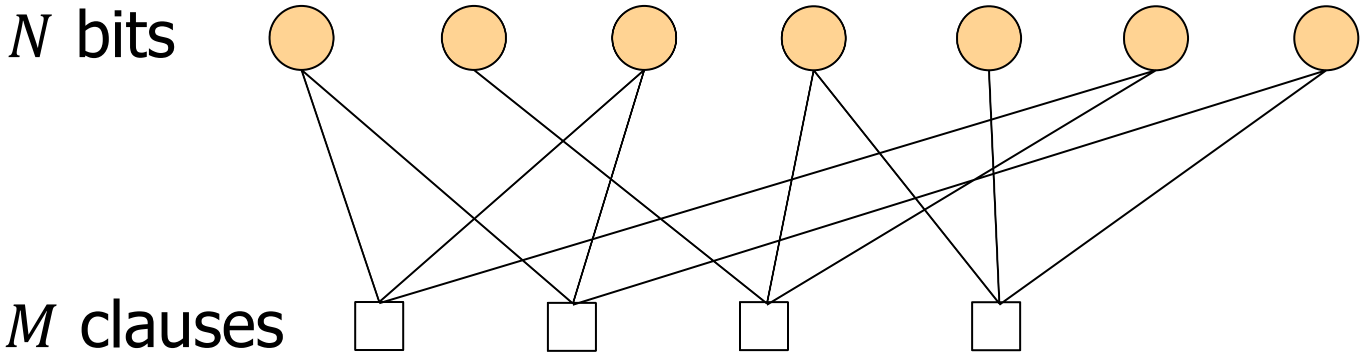
- here, we consider certain "constraint satisfaction" (SAT) problems that are known to be hard classically.
- in SAT problems we ask whether there is an assignment of N bits (or Ising spins) which satisfies all of M clauses (or logical conditions). bits in each clause are chosen at random.

N bits



Satisfiability problems

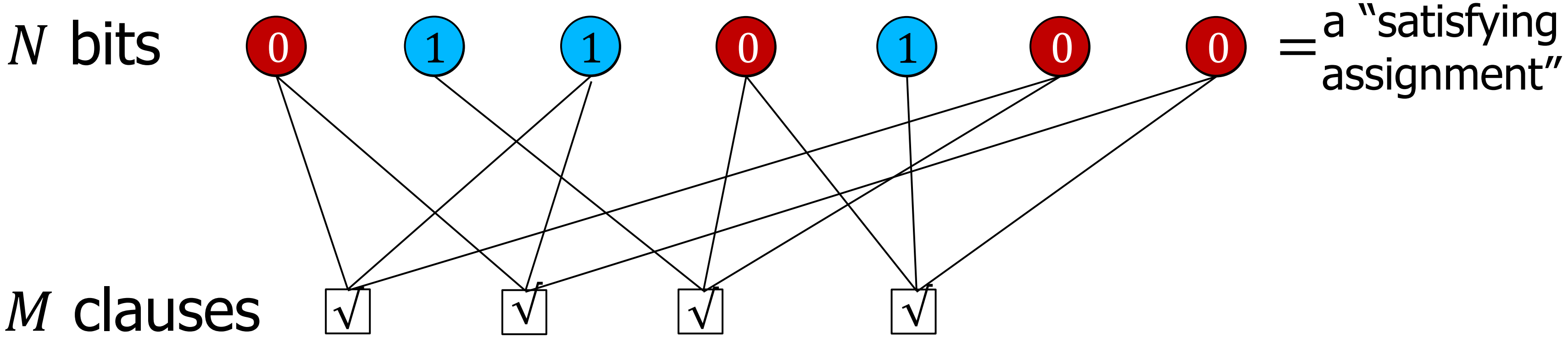
- here, we consider certain "constraint satisfaction" (SAT) problems that are known to be hard classically.
- in SAT problems we ask whether there is an assignment of N bits (or Ising spins) which satisfies all of M clauses (or logical conditions). bits in each clause are chosen at random.
- an example for a clause containing the bits x_1, x_2, x_3 would be: $(x_1 \land \neg x_2 \land \neg x_3) \lor (x_2 \land \neg x_3 \land \neg x_1) \lor (x_3 \land \neg x_1 \land \neg x_2)$.



factor graph

Satisfiability problems

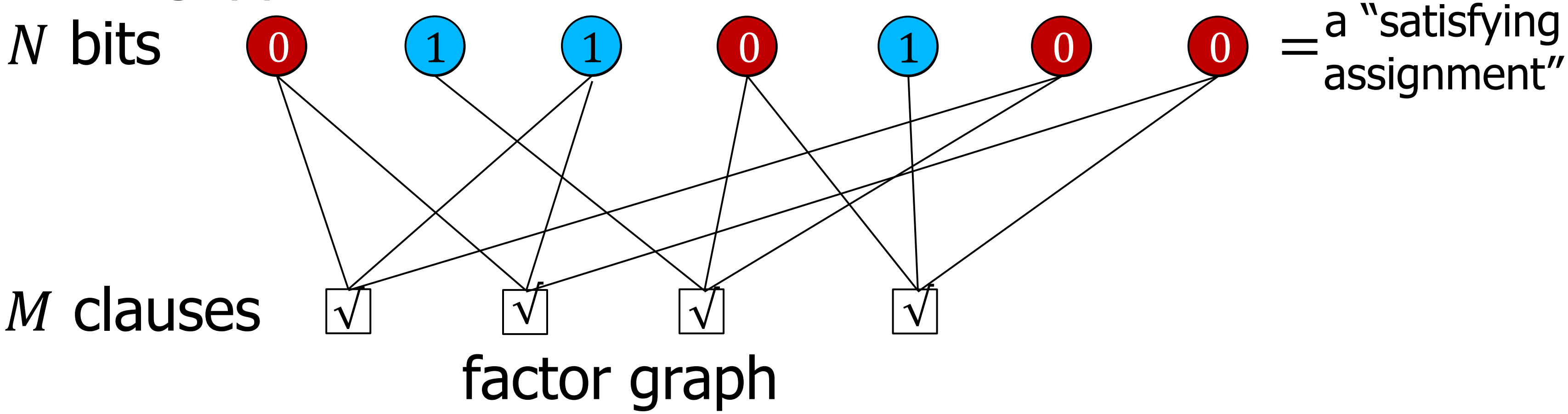
- here, we consider certain "constraint satisfaction" (SAT) problems that are known to be hard classically.
- in SAT problems we ask whether there is an assignment of N bits (or Ising spins) which satisfies all of M clauses (or logical conditions). bits in each clause are chosen at random.
- an example for a clause containing the bits x_1, x_2, x_3 would be: $(x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_2 \wedge \neg x_3 \wedge \neg x_1) \vee (x_3 \wedge \neg x_1 \wedge \neg x_2)$.



factor graph

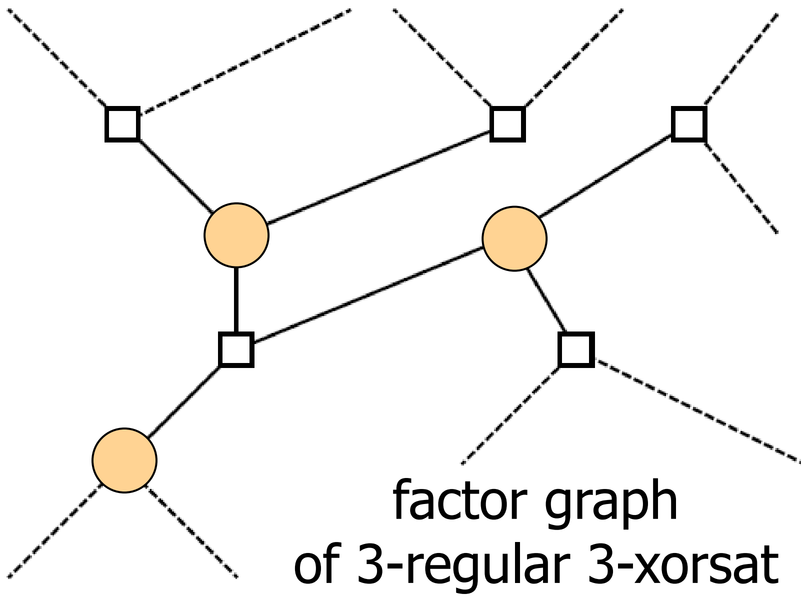
Satisfiability problems

- for small M/N: easy to satisfy all clauses. exponential number of solutions or "satisfying assignments" (SAT phase).
- for large M/N: no satisfying solution exists (UNSAT phase).
- we take the ratio of M/N to be at the *satisfiability threshold*, where it is difficult to find a solution.
- we study random instances with a *unique satisfying assignment* (USA). numerically more convenient because model is gapped.



Satisfiability problems

- the SAT problems we examined:
  - locked 1-in-3: a clause is a triplet of bits picked at random from a pool of N bits. it is satisfied if and only if exactly one bit is 1 and the other two are 0.
  - locked 2-in-4: same as above only with 2 bits out of 4 in each clause that must be 1 [Zdeborová & Mézard, 08].
  - 3-regular 3-xorsat [Jörg et al]: here, each bit is exactly in 3 clauses and a clause is satisfied if the sum of the 3 bits in a clause (mod 2) is a value specified (0 or 1). this problem is in P.



factor graph of 3-regular 3-xorsat

The encoding Hamiltonians

- the SAT problems are encoded in problem Hamiltonians that are sums of clause Hamiltonians, each of the clauses is a sum of products of σ_i^z matrices.

$$\hat{H}_p = \sum_{a=1..M} \hat{H}_a(\sigma_i^z)$$

- the ground state of the problem Hamiltonian $\hat{H}_p$ is a solution to the SAT problem. $\hat{H}_p$ is diagonal in the computational basis.
- we choose the simplest possible driver Hamiltonian (e.g., equal weights):

$$\hat{H}_d = \sum_{i=1..N} \frac{1}{2}(1-\sigma_i^x)$$

- this is a simple transverse-field Hamiltonian. it does not commute with any of the problem Hamiltonians. its ground state is unique and its energy is 0. the gap is 1.

Method

- main goal:

determine the complexity of the QAA for the various SAT problems

- study the dependence of the typical minimum gap

$$\Delta E_{\min} = \min_{s\in(0,1)} \Delta E$$

on the size N (number of bits) of the problem.

- this is because:

$$\mathcal{T} \propto {}^{1}/_{\Delta E^2_{min}}$$

- polynomial dependence → polynomial complexity!

Method

- for each given problem we study several system sizes, because we are interested in size-dependence.
- we consider typically 50 instances per problem size. to obtain "typical behavior" we take medians.
- for each instance, we measure the gap of the system for several values of the parameter s in order to obtain an accurate estimation of the minimum gap.
- for each s value, we run a quantum Monte Carlo (QMC) simulation to obtain the gap numerically (and other measureable quantities).

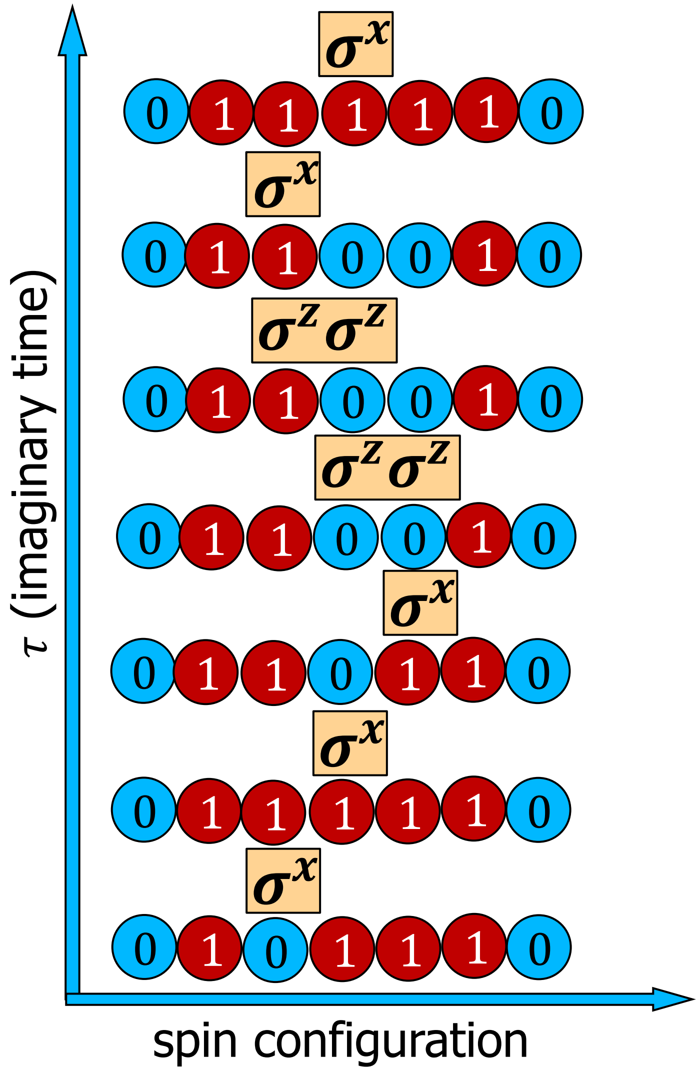
Quantum Monte Carlo

- for large system sizes, we can not use exact diagonalization.
- we employ a continuous-time quantum Monte Carlo (QMC) technique.
- an additional periodic dimension of imaginary time $0 \leq \tau < \beta$. β is the inverse-temperature obeying $\beta \Delta E_1 \gg 1$.
- with QMC we do a sampling of the 2^N states of the Hilbert space. exact-numerical up to statistical errors.
- QMC enables access to the equilibrium properties of the system but also provides indirect access to the system gap.
- here, we basically simulate spin-$1/2$ systems with different interactions and different sizes.
- we employ parallel tempering (swapping configurations of adjacent s values) which speeds up equilibration.

Quantum Monte Carlo: SSE

- we use the stochastic series expansion (SSE) algorithm devised by Anders Sandvik [Sandvik, 1991,1992,1994].
- algorithm is based on a Taylor series expansion of the partition function $Z = Tr\left[e^{-\beta \hat{H}}\right]$. no systematic errors.
- algorithm enables both local and global (cluster or loop) updates which in most cases prove to be more efficient than single-spin-flip (local) updates.

a typical segment of an SSE configuration



Extracting the gap

- we extract the gap of the system by measuring and analyzing different-imaginary-time correlation functions of certain operators:

$$C_A(\tau) = \langle \hat{A}(\tau)\hat{A}(0)\rangle - \langle \hat{A}\rangle^2$$

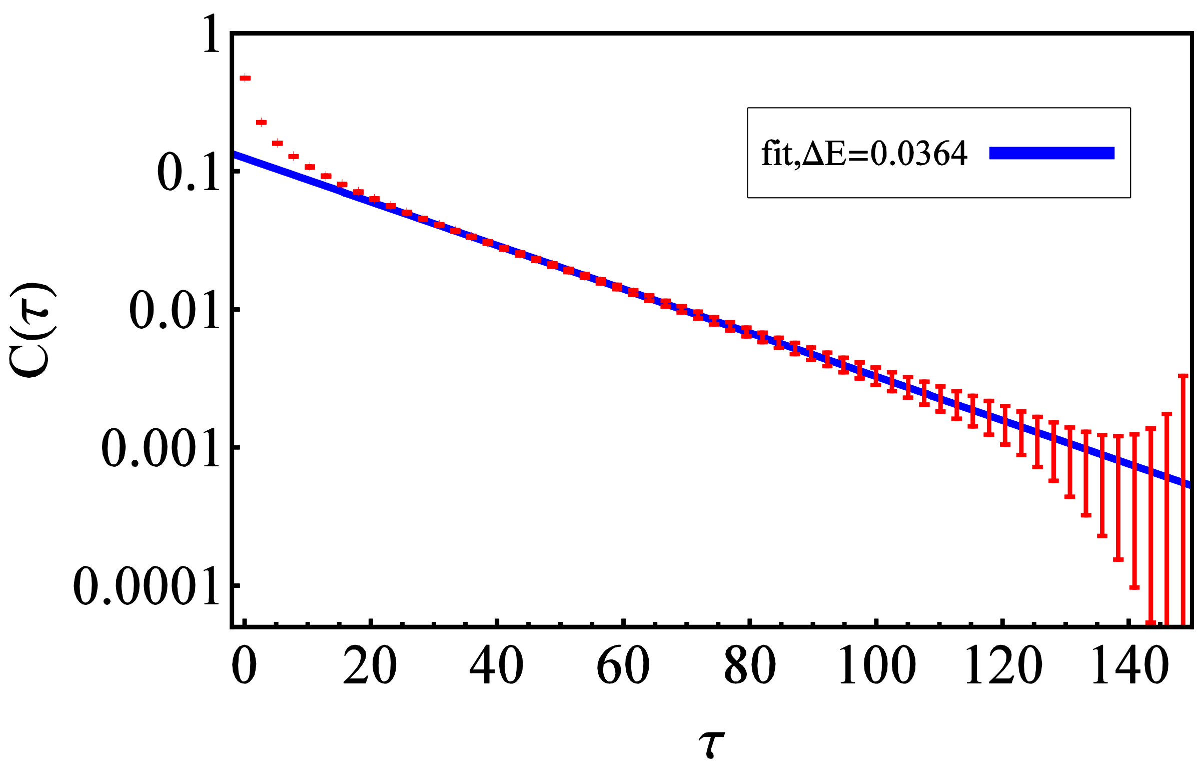
- if $\beta \Delta E_1 \gg 1$ (system is in its ground state) then at long imaginary times we have:

$$C_A(\tau) \cong \left|\langle 0|\hat{A}|1\rangle\right|^2 e^{-\Delta E_1 \tau}$$

- i.e., only the slowest-decaying exponent survives (provided that the corresponding matrix element does not vanish).

Extracting the gap

- if conditions are right, then it is possible we extract the gap of the system.
- we use a straight line fit on a log-linear scale.
- however, data becomes very noisy at large imaginary times.



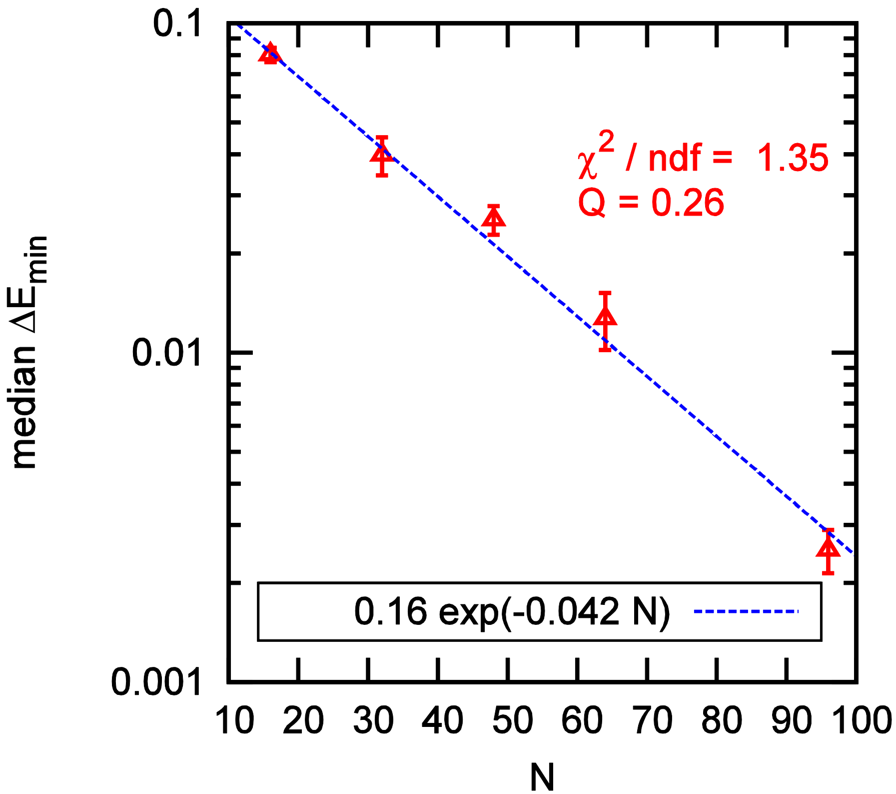
correlation function of a 64-spin instance of the locked-1-in-3 problem ($s = 0.39, \beta = 1024$). log-linear scale.

QMC results

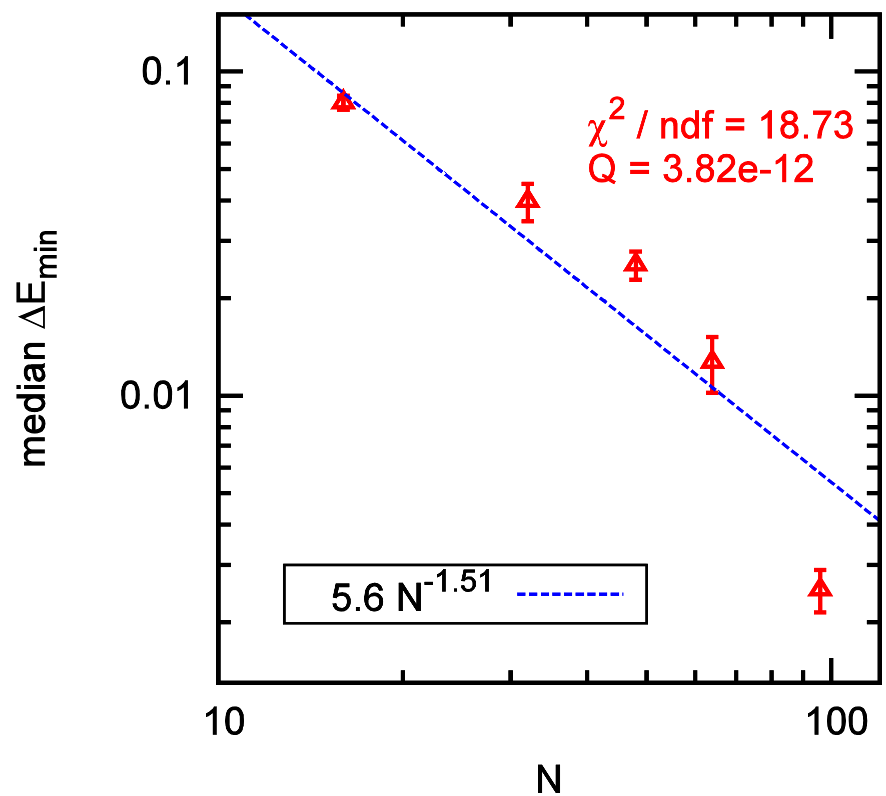
- main goal: to determine the complexity of the QAA for the various SAT problems.
- for each of the problems studied, we look at the dependence of the typical (median) minimum gap on the size of the problem.
- a polynomially decreasing gap would mean a polynomially increasing running time and hence QAA could be called efficient.
- an exponentially decreasing gap would mean that the QAA is not more efficient than the best classical algorithm.
- heavy QMC simulations. hundreds/thousands of cores, running in some cases for weeks/months.

Locked 1-in-3 SAT

plots of the median minimum gap vs problem size N



exponential (log-linear) fit

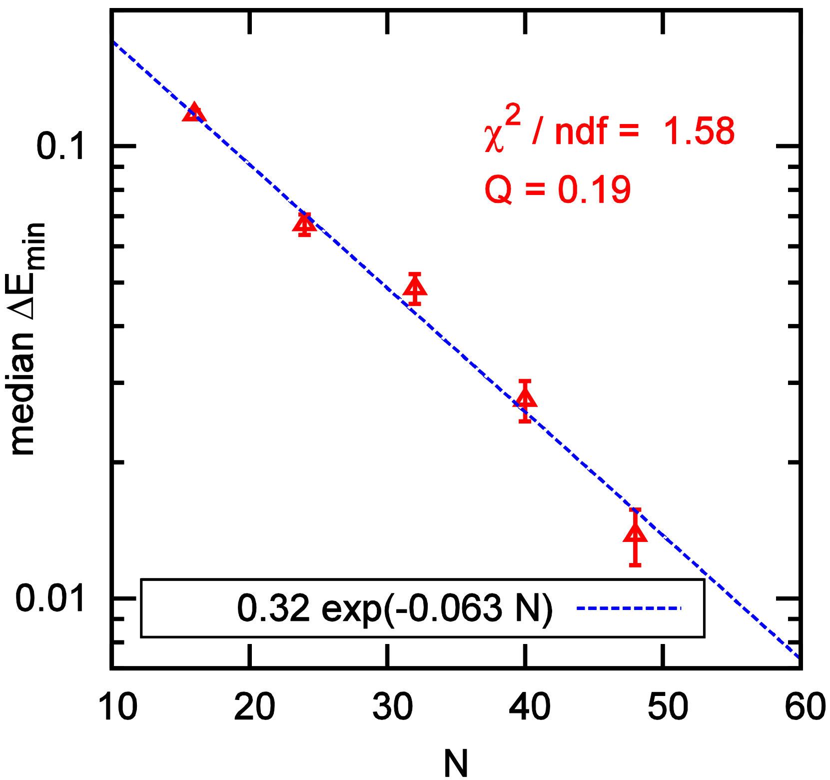


power-law (log-log) fit

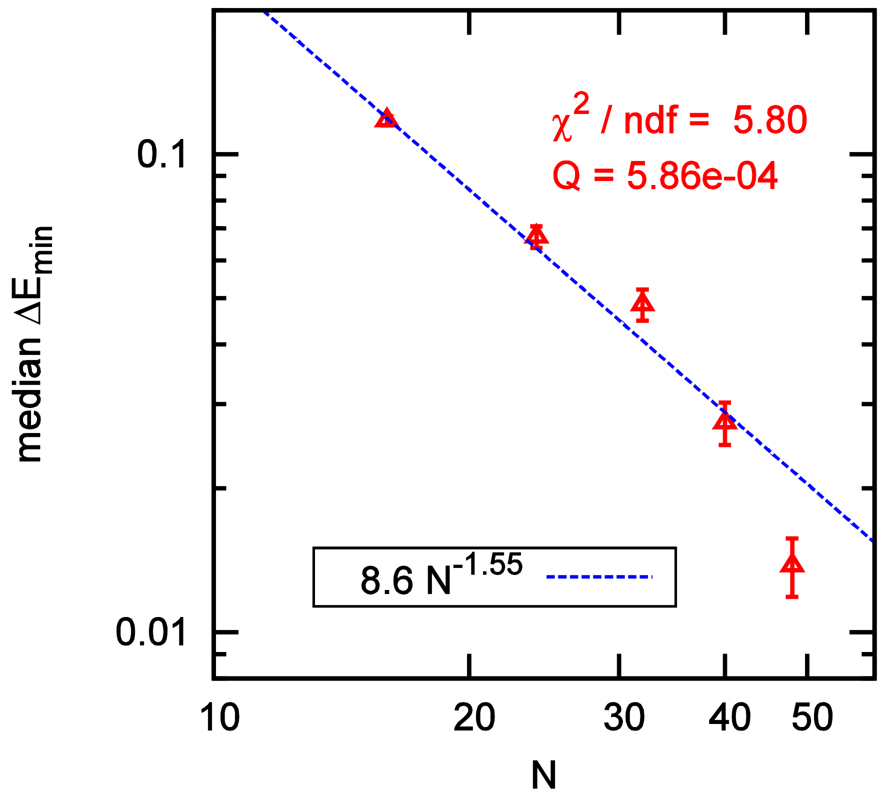
clearly, the behavior of the minimum gap is exponential.

Locked 2-in-4 SAT

plots of the median minimum gap vs problem size N



exponential (log-linear) fit

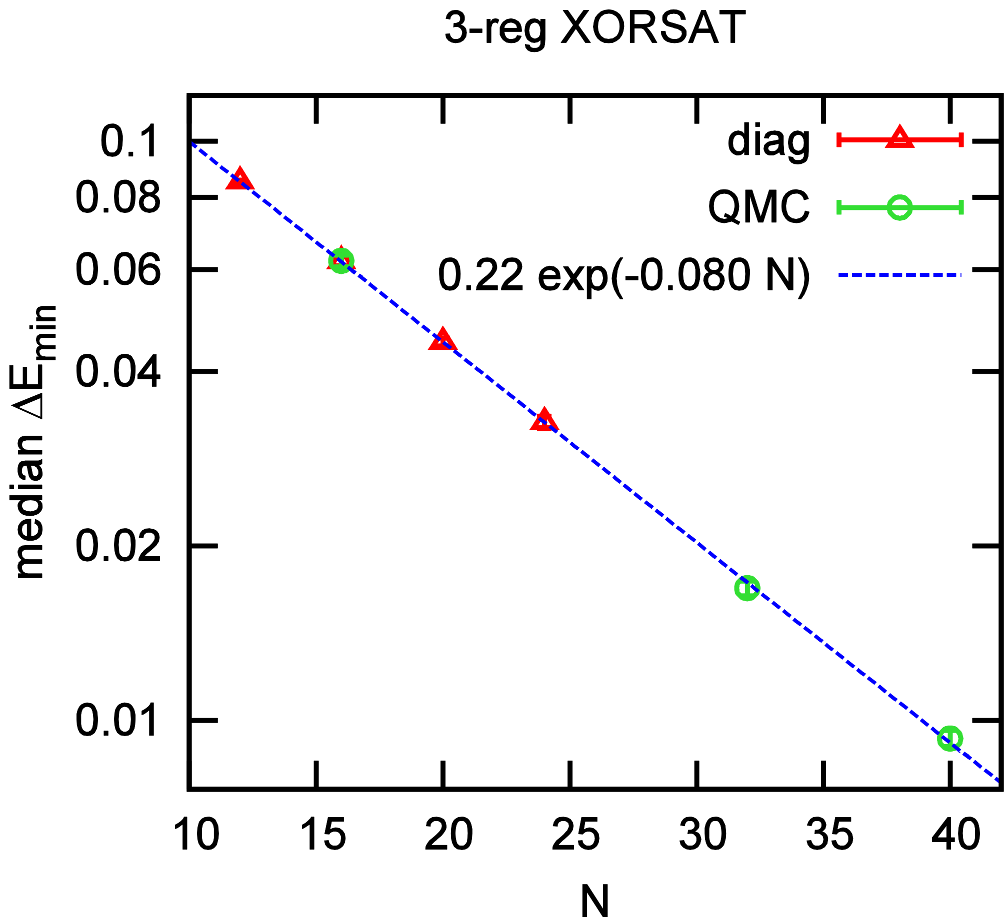


power-law (log-log) fit

clearly, the behavior of the minimum gap is exponential.

3-regular 3-XORSAT

exponential (i.e., log-linear) plot of the median minimum gap



median minimum gap is exponential,
even from small N, and even though problem is in P.

Comparison with a classical algorithm

- WalkSAT is a classical, heuristic, local search algorithm.
- it is a reasonable classical algorithm to compare with QAA [Guidetti and Young, 2010].
- the algorithm itself is very simple:
  - pick at random an unsatisfied clause and flip a bit in that clause.
  - with some probability this bit is chosen to be the one which causes the fewest previously satisfied clauses to become unsatisfied, and otherwise it is chosen at random.
  - repeat until the number of unsatisfied clauses is zero.

Comparison with a classical algorithm

- WalkSAT is a classical, heuristic, local search algorithm.
- it is a reasonable classical algorithm to compare with QAA [Guidetti and Young, 2010].
- the complexity of WalkSAT is determined by the amount of "bit flips" the algorithm performs until it reaches a solution.

$$\mathcal{T} \propto N_{\text{flips}} \sim \exp[\mu N]$$

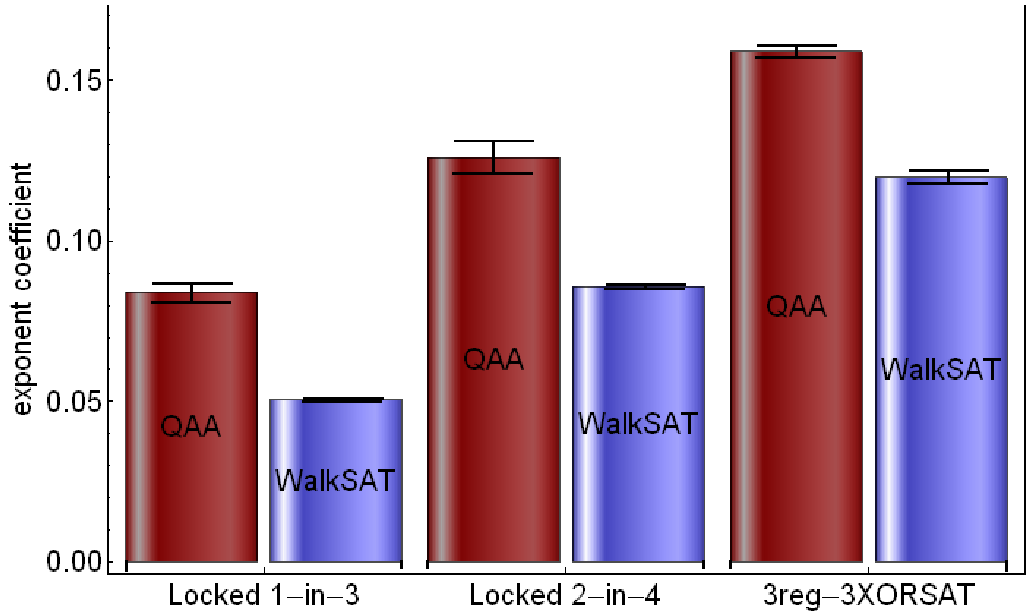
- for the QAA, we have

$$\mathcal{T} \propto \exp[2cN] \quad \text{for} \quad \Delta E_1 \propto \exp[-cN]$$

- we can therefore compare exponent coefficients.

Comparison with a classical algorithm

- running times are proportional to $\exp[\mu N]$ where N is system size.



a comparison of the μ values among the different models and between QAA and WalkSAT

- WalkSAT is better, however we see the same trend.

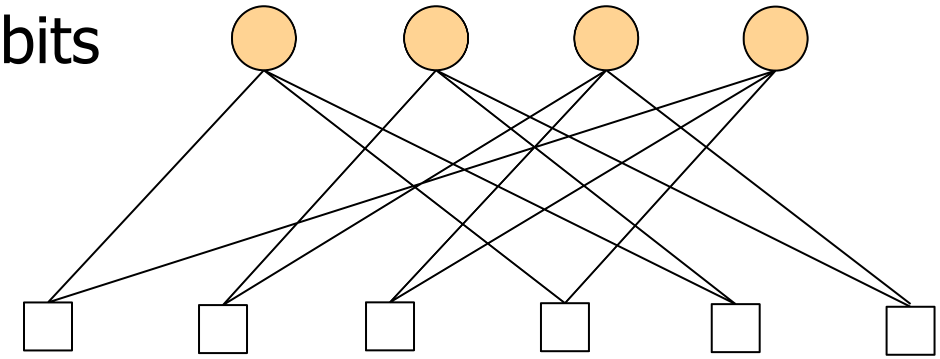
- important to remember: we used here the simplest implementation of the QAA for instances with USA. algorithm can certainly be improved.

3-regular random antiferromagnet (3-reg Max-Cut)

3-regular Max-Cut

- we have also studied one "MAX" (i.e., optimization) problem.
- MAX means that we are in the UNSAT phase, and would like to find the configuration with the least number of unsatisfied clauses.
- 3-regular: each bit is in exactly 3 clauses.
- Max-Cut: sum of two bits (product of two spins) must be a specified value.

N bits



$M = 3N/2$ clauses

factor graph of the 3-regular Max-Cut

3-regular Max-Cut

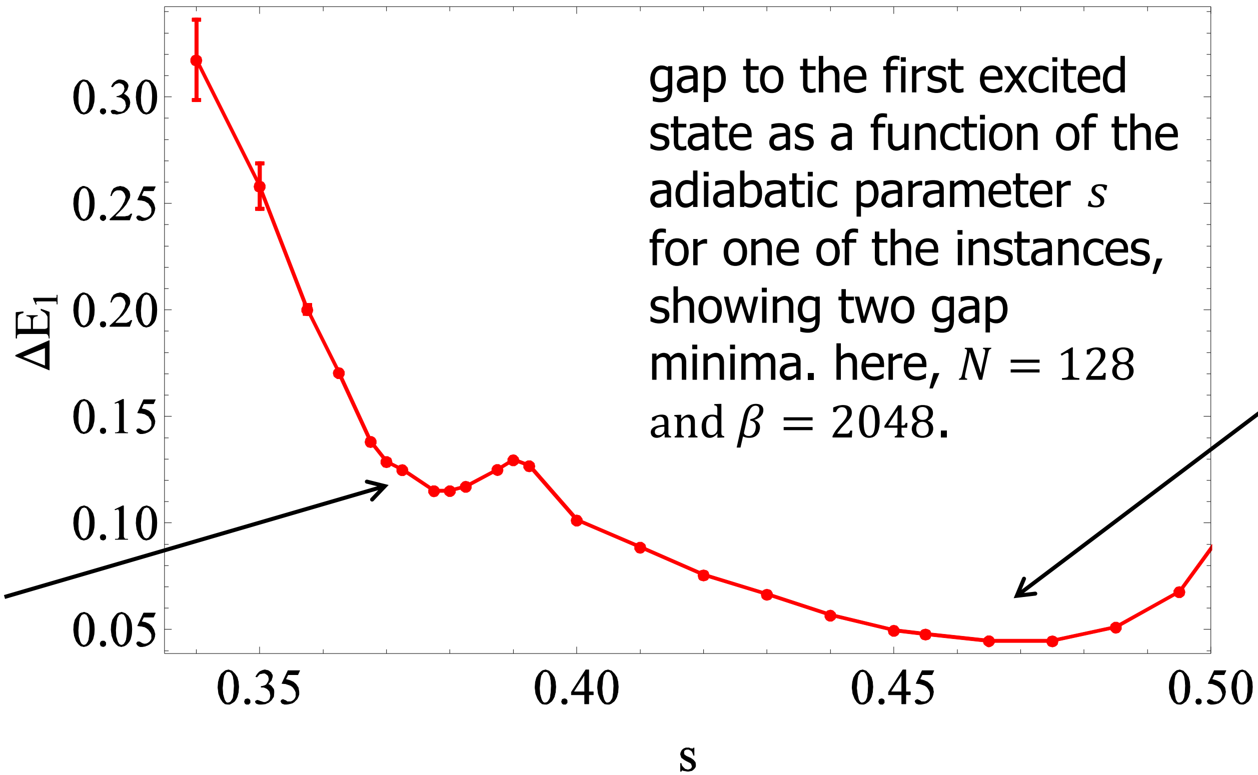
- in our case, the Hamiltonian of a clause is:

$$\hat{H}_a = \frac{1}{2}\left(\sigma^z_{a1}\sigma^z_{a2} + 1\right)$$

- product of spins in a clause must be -1 to satisfy the clause.
- this is a 3-regular antiferromagnet on a random graph. note the symmetry under bit flips.
- however, solution is not a simple "up-down" antiferromagnet because of loops of odd length. in fact, this is a *spin-glass*.
- after adding a Driver Hamiltonian, there is a quantum phase transition above which symmetry is spontaneously broken.
- "Cavity" calculations (Gosset/Zamponi) find the transition at $s \approx 0.36$.

3-regular Max-Cut

- we observe minima near the expected phase transition (where the critical point was determined precisely)
- there are however additional avoided crossings inside the spin-glass phase as well

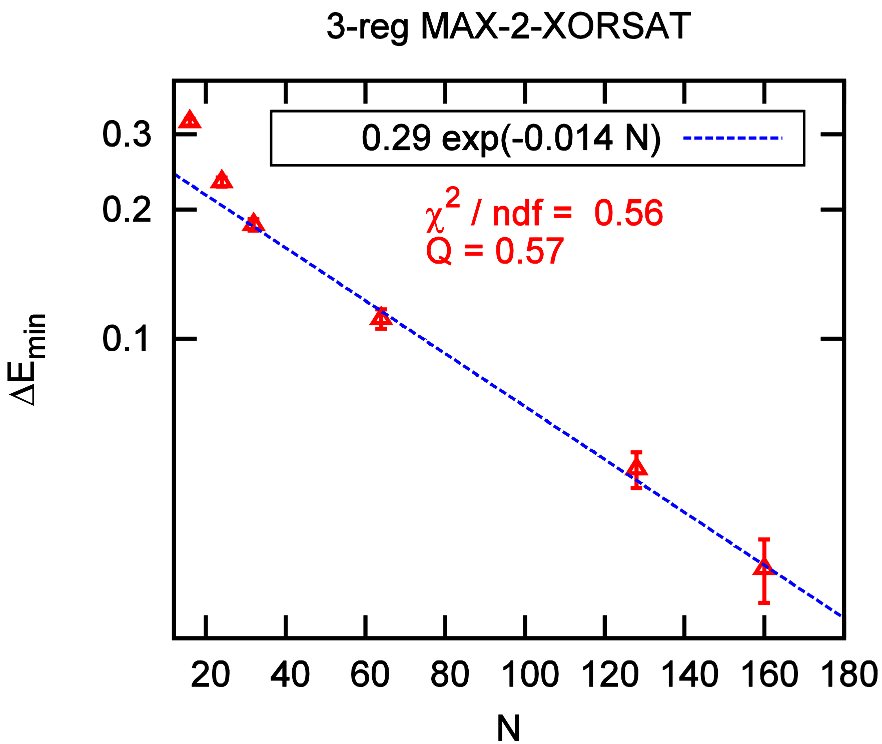


phase transition minimum

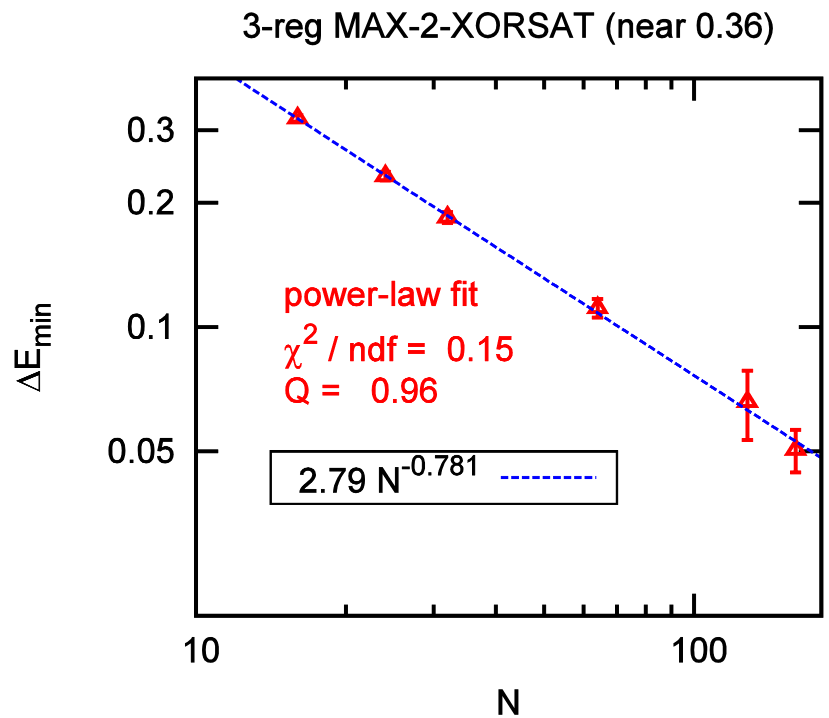
additional minimum inside the spin-glass phase

3-regular Max-Cut

median minimum gap in the vicinity of the quantum transition



exponential (log-linear) fit



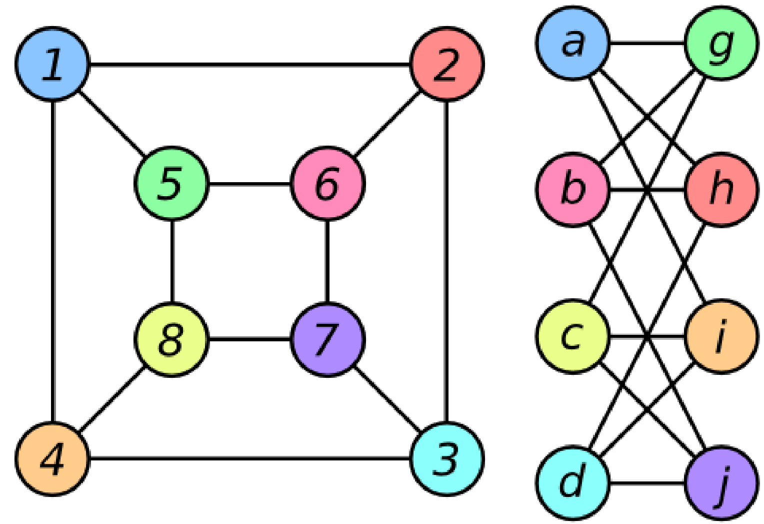
power-law (log-log) fit near phase transition

gap is polynomial near the phase transition, however additional avoided level crossings lead to an exponential gap

The graph isomorphism problem

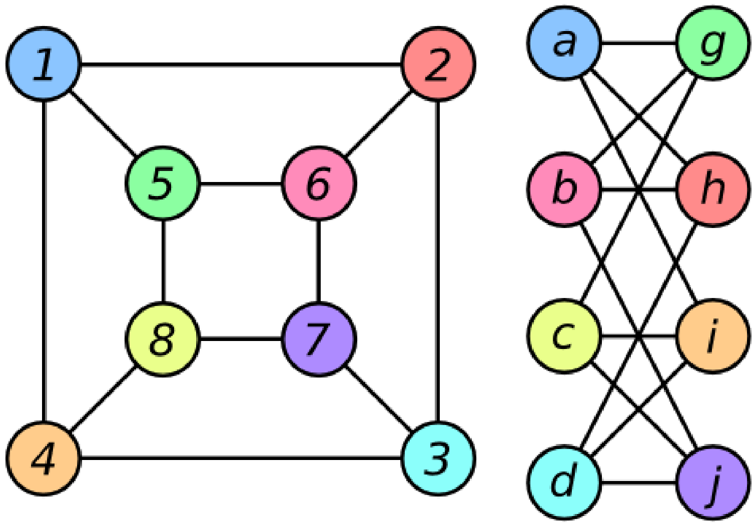
The graph isomorphism problem

- are two graphs the same upon permuting the indices?
- how could one use adiabatic quantum computation to answer this question?



- conjecture: all non-isomorphic graphs can be distinguished by putting a suitable Hamiltonian on the edges of a graph:
  - we construct a problem Hamiltonian for each graph.
  - we run the QAA a multiple number of times.
  - we compute appropriate average physically measurable quantities by repeated measurements.

The graph isomorphism problem



- if the Hamiltonian and the quantities we choose are invariant under permutation of the indices, isomorphic graphs will give the same results.
- we hypothesize that non-isomorphic graphs can always be distnguished.
- we have tested the hypothesis for some small graphs $(N \leq 29)$ from various families of graphs that are known to be hard to distinguish (same adjacency matrices).
- so far, method seems to work if measurements are accurate enough.

The graph isomorphism problem

- we tried the "spin-glass" antiferromagnet on the graph:

$$\hat{H}_p(G)=\sum_{\langle i,j\rangle\in G}\sigma_i^z\sigma_j^z$$

- main results are for Strongly Regular Graphs (SRG's; families of similar but non-isomorphic graphs) but not just. we considered sizes from $N = 15$ to 29 vertices.

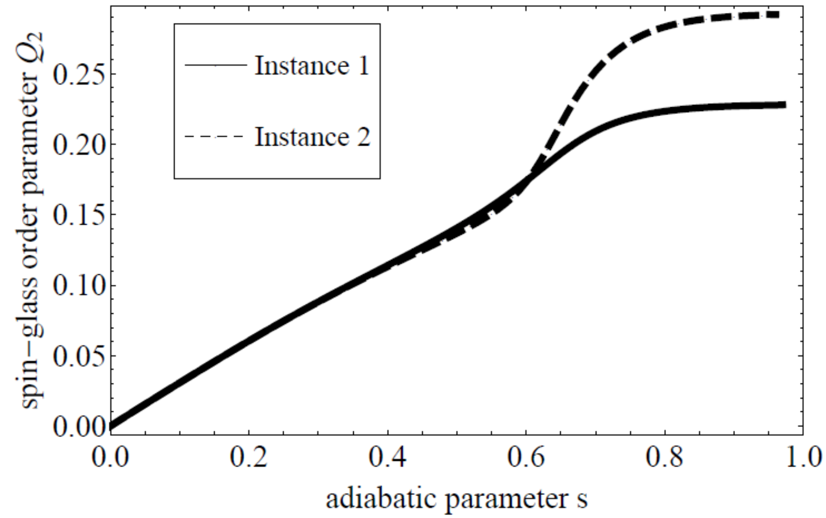
- we computed energy, x-magnetization (M_x) and the spin glass order parameter (Q_2) for different values of the adiabatic parameter s:

$$M_x = \frac{1}{N}\sum_{i=1}^{N}\langle\sigma_i^x\rangle$$

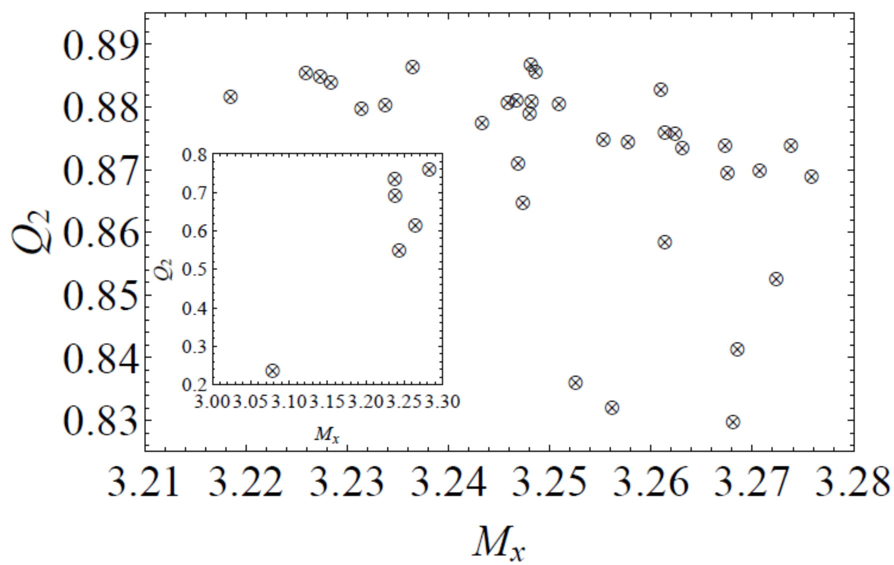
$$Q_2 = \sqrt{\frac{1}{N(N-1)}\sum_{i\neq j}^{N}\langle\sigma_i^z\sigma_j^z\rangle}$$

The graph isomorphism problem

- the value of Q_2, the spin-glass order parameter, in the ground state for the two non-isomorphic SRG's on $N = 16$ vertices, as a function of the adiabatic parameter s. the two graphs are clearly distinguished.



- scatterplot of Q_2 against M_x in the $s \to 1$ limit for the 41 SRG's with $N = 29$. the QAA distinguishes all graphs in the family in that limit (although some of the values are close together).



The graph isomorphism problem

- perhaps there are “better” Hamiltonians than the one chosen here. here, we have used “glassiness” to solve the graph isomorphism problem.
- perhaps there are better measurements that can be performed in order to distinguish between graphs, e.g., susceptibilities. here, we have mainly used the spin-glass order parameter.
- can be tested experimentally on existing D-Wave hardware with relatively minor modifications.
- it is unclear whether or not the algorithm is efficient. what is the nature of the quantum phase transition? need to investigate size-dependence of minimum gap.
- clearly more testing is needed.

Conclusions and future research

Conclusions

- for the SAT problems investigated, we don't find that QAA is better than state-of-the-art classical algorithms.
- we find that the harder a problem is for classical algorithms (WalkSAT), the harder it is also for the QAA.
- for the Max-Cut (random antiferromagnet) problem, results point to a polynomially decreasing gap near the quantum phase transition. it seems however that the overall gap behavior is exponential.
- QAA seems to be able to solve the graph isomorphism problem (more tests are needed) however the efficiency of the algorithm is not yet known.