Solutions to systems of equations from adiabatic quantum computing

Chia Cheng Chang

Interdisciplinary Theoretical and Mathematical Sciences Program (iTHEMS), RIKEN

University of California - Berkeley

Lawrence Berkeley National Lab

In collaboration with

Arjun Gambhir

University of California - Berkeley

Travis Humble

Oak Ridge National Lab

Shigetoshi Sota

Center for Computational Sciences (CCS), RIKEN

Qubits North America 2018

Knoxville, Tennessee

Systems of equations

Solutions to systems of equations are ubiquitous in science, engineering and mathematics.

Problem definition

Linear system of equations

index notation

$$P_{ij}^{(1)} x_j = P_i^{(0)}$$

matrix notation

$$\begin{pmatrix} P_{00}^{(1)} & P_{01}^{(1)} \\ P_{10}^{(1)} & P_{11}^{(1)} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} P_0^{(0)} \\ P_1^{(0)} \end{pmatrix}$$

Polynomial system of equations

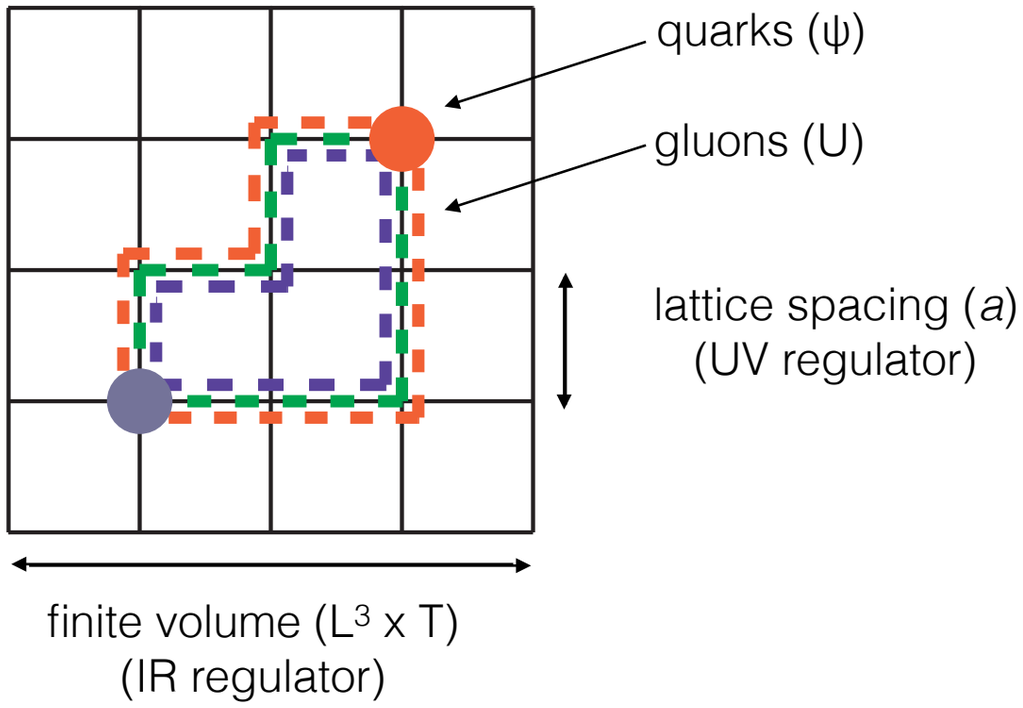
$$P_{ij}^{(1)} x_j + P_{ijk}^{(2)} x_j x_k + \cdots = P_i^{(0)}$$

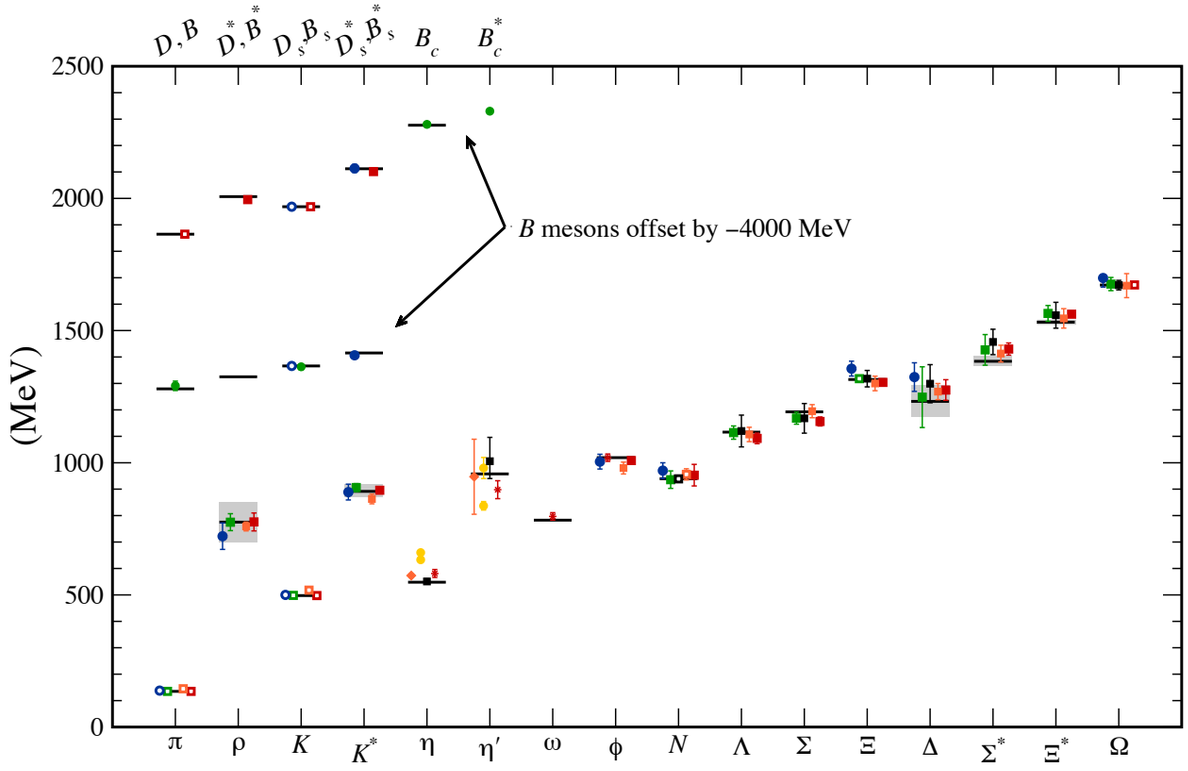
High-dimensional function spaces are challenging for classical algorithms.

Can quantum computing tackle this?

Lattice QCD application

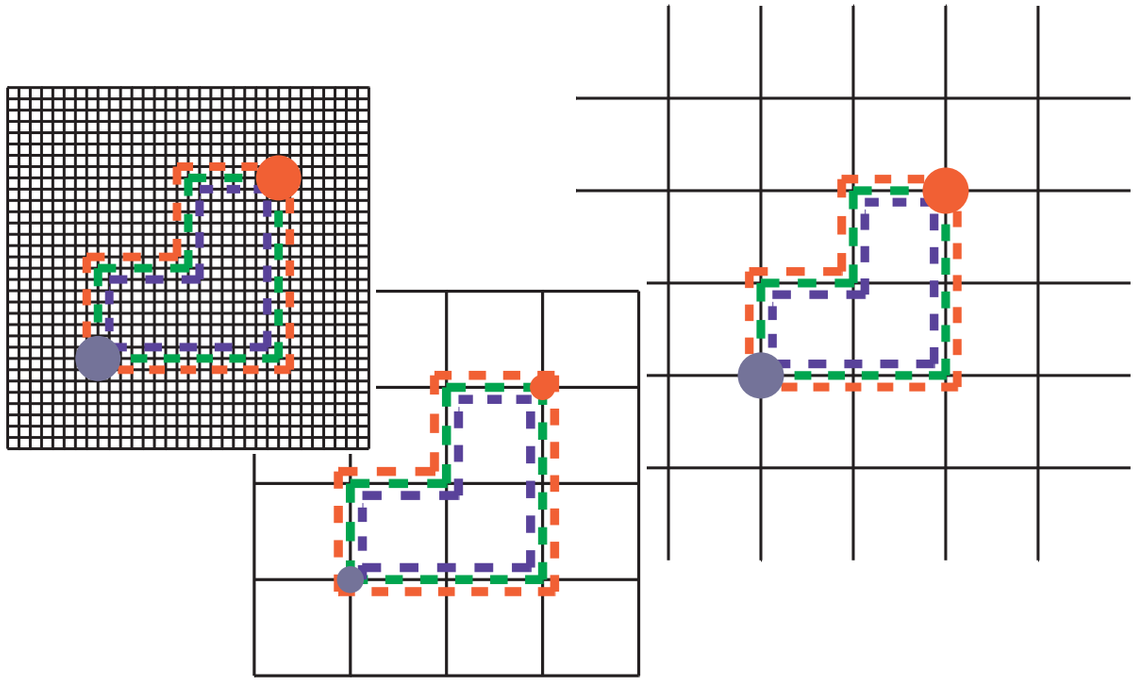
Quantum chromodynamics (QCD) is our modern description of the strong interaction.
After setting ~3 input parameters, all of nuclear physics is a prediction!





Major lattice uncertainties and related issues

- continuum limit $t_{\text{comp.}} \propto 1/a^6$
- infinite volume $t_{\text{comp.}} \propto V^{5/4}$
- light pion mass condition number ~$1/m_\pi$
  exponentially bad
  - signal-to-noise



Lattice QCD application

System of linear equations in Lattice QCD

On the lattice, we simulate a small 4 dimensional piece of the universe. (~ 5 protons wide).

This translates to approximately 1,000,000 to 1,000,000,000 degrees of freedom
(space, time, and quantum degrees of freedom)

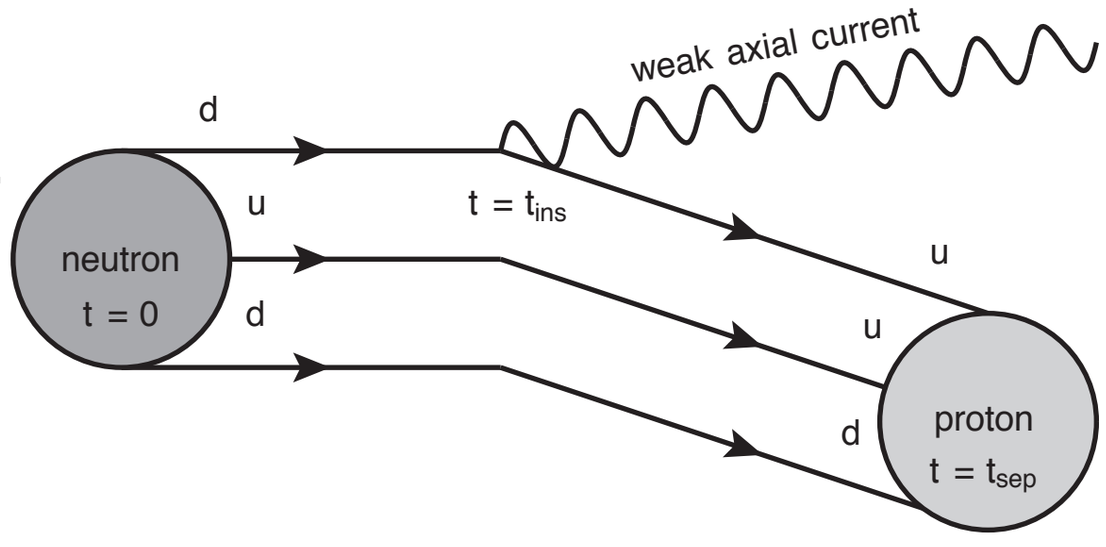
Solving the Dirac equation is a linear problem.
(Equation of motion for quarks)

$$\not{D}\psi = S$$

$$P^{(1)}x = P^{(0)}$$

where the Dirac operator is a rank 10^6—10^9 matrix.

Neutron decay Feynman diagram



Scientific impact
Lattice field theory + computational toolkit allows
for a fundamental understanding of our universe.

Neutron decay published May 2018 in Nature.
Nature 558, 91-94 (2018)

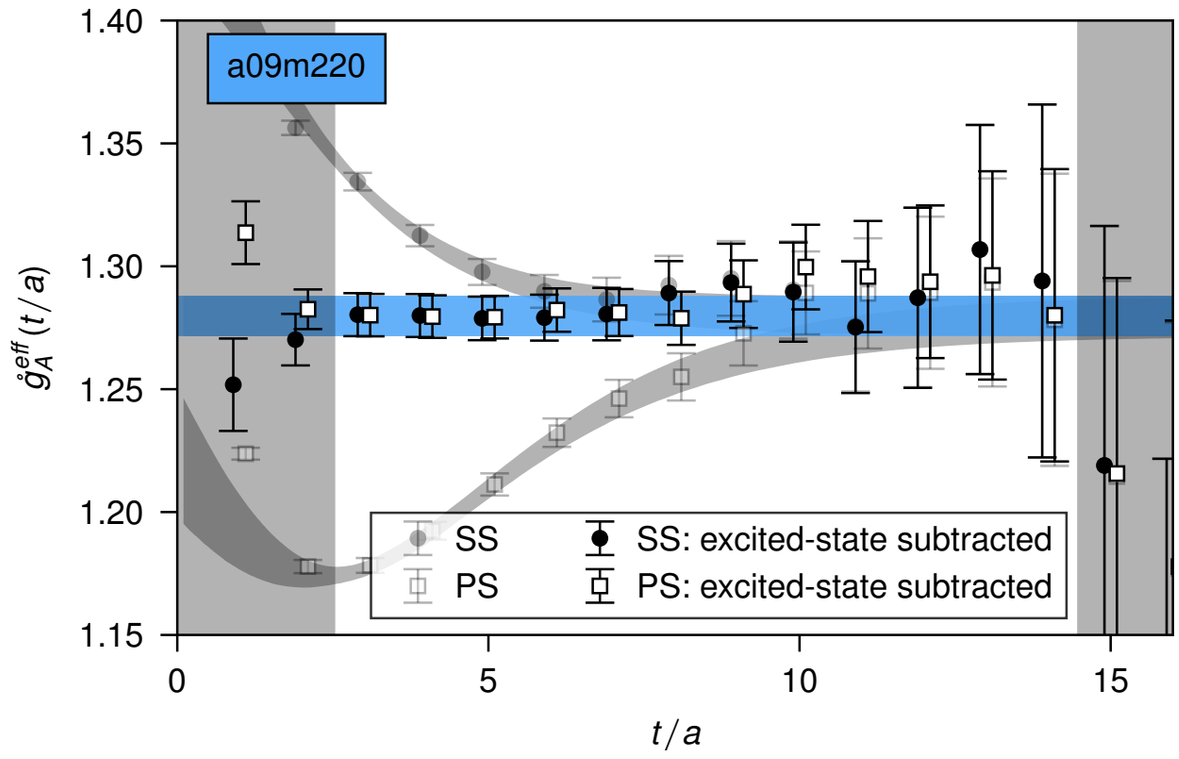
At ~1 cent / cpu-hour, leadership-class supercomputing calculations costs millions to run.

Least squares minimization

Regression is a broadly applicable technique to extract parametric information from data.

Very important part of a lattice QCD calculation.

LQCD calculations come with many artifacts that need to be removed by regression.



Linear and polynomial regression can be mapped to a system of equations.

Correlated least-squares loss function

$$\chi^2 \equiv [F(x,\{p\}) - \langle y \rangle]_i \, S_{ij}^{-1} \, [F(x,\{p\}) - \langle y \rangle]_j$$

Fit function

$$F(x_i,\{p\}) = \sum_{n=1}^{P} p_n f_n(x_i)$$

Minimizing loss function (assuming linear) yields a linear system of equations

$$P^{(1)} = \begin{pmatrix} f_0(x_i)S_{ij}^{-1}f_0(x_j) & \dots & f_0(x_i)S_{ij}^{-1}f_P(x_j) \\ \vdots & \ddots & \vdots \\ f_P(x_i)S_{ij}^{-1}f_0(x_j) & \dots & f_P(x_i)S_{ij}^{-1}f_P(x_j) \end{pmatrix} \quad P^{(0)} = \begin{pmatrix} f_0(x_i)S_{ij}^{-1}y_j \\ \vdots \\ f_P(x_i)S_{ij}^{-1}y_j \end{pmatrix} \quad P^{(1)}x = P^{(0)}$$

Classical algorithms for linear equations

Direct method

Gauss Jordan elimination (solve for inverse of matrix)

- Computational complexity $\mathrm{O}(n^3)$
- Intractable for large linear systems (*e.g.* LQCD)

$$\begin{pmatrix} P_{00}^{(1)} & P_{01}^{(1)} \\ P_{10}^{(1)} & P_{11}^{(1)} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} P_0^{(0)} \\ P_1^{(0)} \end{pmatrix}$$

Iterative methods

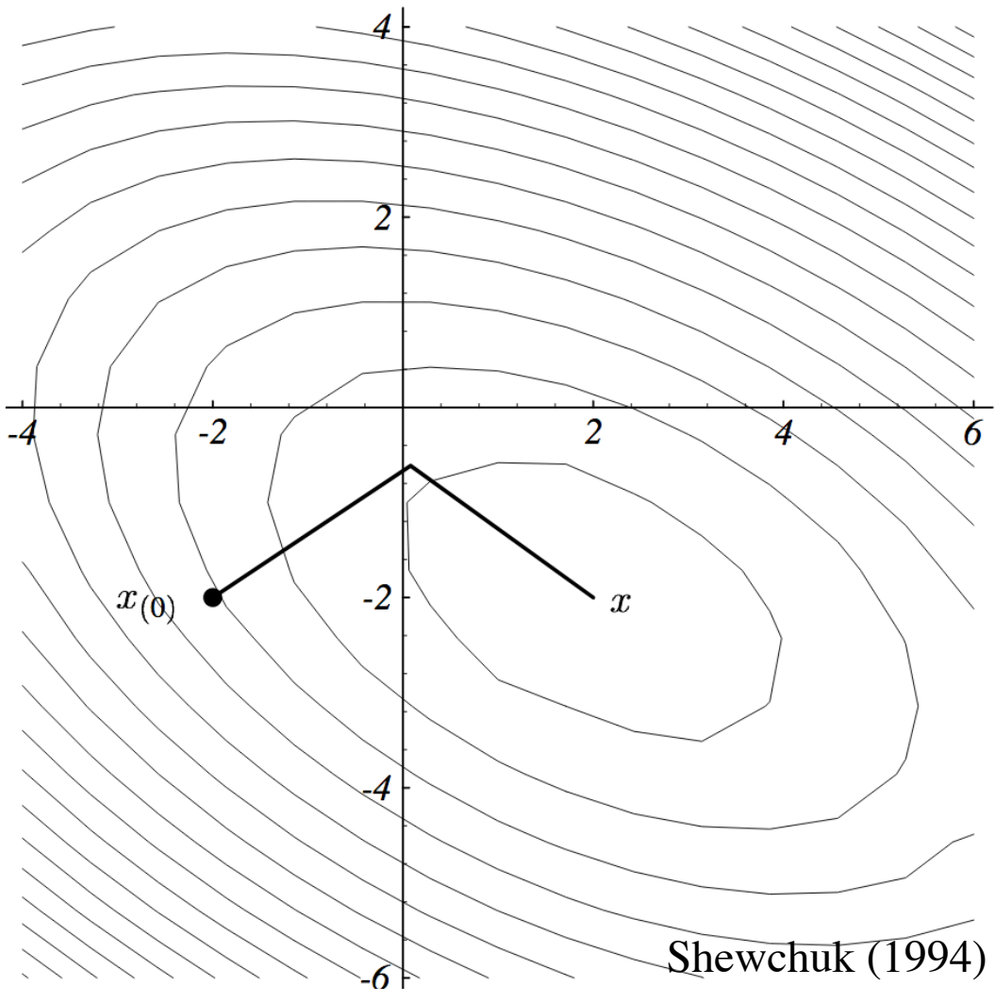
Conjugate-gradient

- Works only on symmetric positive-definite matrix.
  - Solve normal equation: $P^{(1)\dagger} P^{(1)} x = P^{(1)\dagger} P^{(0)}$
- Builds basis with conjugate vectors.

- First entry is the negative gradient of

$$\frac{1}{2} x^T P^{(1)} x - x^T P^{(0)}$$

- Conjugate directions are minimized afterwards.
- Computational complexity $\mathrm{O}(n^2)$per iteration.
  - Iterations as square root of condition number.



Shewchuk (1994)

Preconditioning

- Solve $M^{-1} P^{(1)} x = M^{-1} P^{(0)}$
- M^{-1} is chosen to lower condition number of $P^{(1)}$ by leveraging details of $P^{(1)}$

Ising model and adiabatic quantum computing

n - body Ising Hamiltonian (some more notation and definitions)

$$H^{\text{Ising}}(\psi) = \sum_i h_i \psi_i + \sum_{\langle i,j \rangle} J_{ij} \psi_i \psi_j + \sum_{\langle i,j,k \rangle} K_{ijk} \psi_i \psi_j \psi_k + ...$$

ext. mag. 2-body 3-body

The sum in brackets denote nearest neighbors.

- An all-sites to all-sites interacting Ising model has infinite dimensions.

Adiabatic evolution

$$H(s) = (s-1)H^{\text{init}} + sH^{\text{problem}} \qquad H^{\text{init}} = C_i \psi_{x,i}$$

Initiate a spin-up state in *x*, which is fully entangled in *z* and adiabatically evolve in *s*

Adiabatic condition depends on energy gap of *time-dependent* Hamiltonian

$$T \geq \text{O}\left(\frac{\left|\left| H^{\text{init}} - H^{\text{problem}} \right|\right|^2}{\epsilon \min_{s \in [0,1]} \Delta(H(s))^3} \right)$$

Adiabatic quantum computing can solve problems that can be mapped to the Ising Model.

Mapping systems of equations to Ising model

(This is the only important slide in this whole talk…)

Rewrite system of equations $P_{ij}^{(1)}x_j + P_{ijk}^{(2)}x_j x_k + \cdots = P_i^{(0)}$

into a minimization problem $\mathrm{Min}\left[\frac{1}{2}P_{ij}^{(1)}x_i x_j + \frac{1}{3}P_{ijk}^{(2)}x_i x_j x_k + \ldots - P_i^{(0)}x_i\right]$

Map *R*-spin chain to a superposition of 2^R decimal numbers

$$x_i = a_i \sum_{r=0}^{R-1} 2^r \psi_{r,i} + b_i \qquad \text{where} \qquad \psi \in [0,1] \quad \text{(QUBO definition)}$$

For a system of *N* linear equations the mapping is explicitly

$$\begin{aligned} H^{\mathrm{QUBO}}(\psi) = &\begin{pmatrix} \psi_{1,1} \\ \vdots \\ \psi_{R,N} \end{pmatrix}^T \left[\begin{pmatrix} a_1^2 P_{11}^{(1)} & \cdots & a_1 a_N P_{1N}^{(1)} \\ \vdots & \ddots & \vdots \\ a_N a_1 P_{N1}^{(1)} & \cdots & a_N^2 P_{NN}^{(1)} \end{pmatrix} \otimes \begin{pmatrix} 2^0 2^0 & \cdots & 2^0 2^{R-1} \\ \vdots & \ddots & \vdots \\ 2^{R-1} 2^0 & \cdots & 2^{R-1} 2^{R-1} \end{pmatrix} \right. \\ &\left. -2 \begin{pmatrix} a_1\left(P_1^{(0)} - b_1 \sum_i P_{1i}^{(1)}\right) & & \\ & \ddots & \\ & & a_N\left(P_N^{(0)} - b_N \sum_i P_{Ni}^{(1)}\right) \end{pmatrix} \otimes \begin{pmatrix} 2^0 & & \\ & \ddots & \\ & & 2^{R-1} \end{pmatrix} \right] \begin{pmatrix} \psi_{1,1} \\ \vdots \\ \psi_{R,N} \end{pmatrix} \\ = &\Psi_i Q_{ij} \Psi_j \end{aligned}$$

In principle a direct solve (global minimum) independent of condition number or sparseness.
Quantum computers can simultaneously evaluate an exponential number of solutions.
Note: PRL 103 150502 (2009) (HHL algorithm) outlines linear solver for gate-model QC.

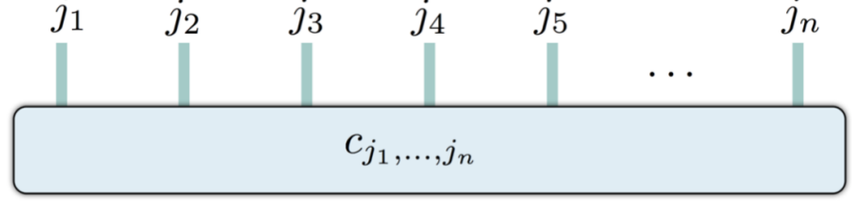
Classical Ising model solvers

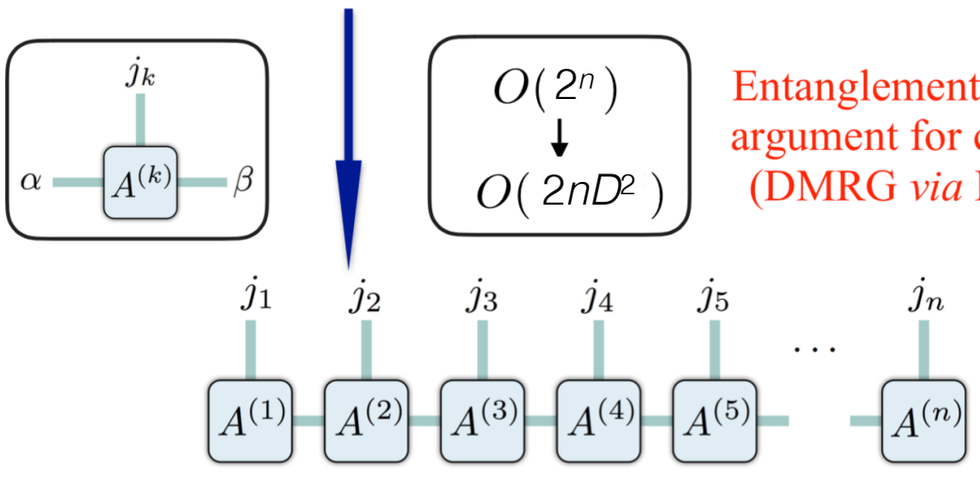
Brute force solution — try all 2^n solutions! (works for super small systems)

Density matrix renormalization group (DMRG)

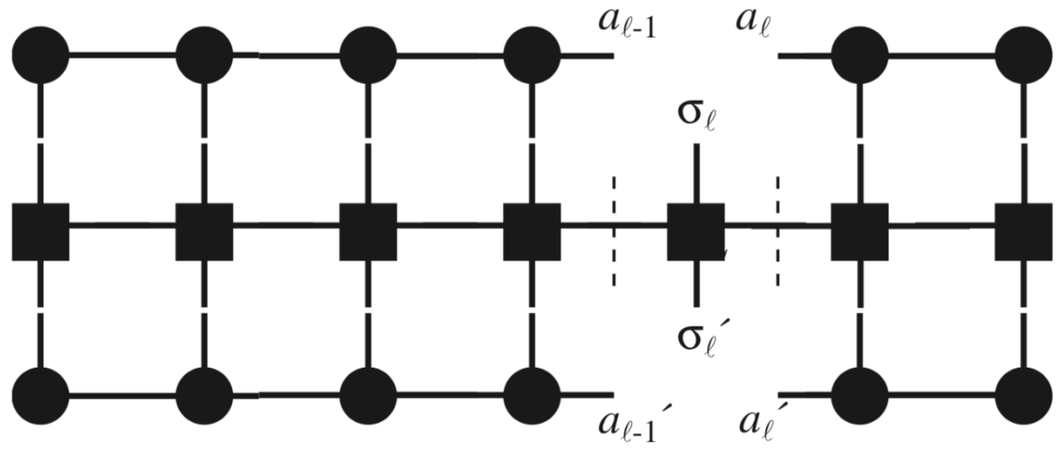
System of *n* spins means diagonalizing a 2^n x 2^n matrix to solve for spectrum (impossible).
In practice a large part of the space can be truncated.
DMRG keeps states with largest entanglement only (in contrast Wilson is energy cutoff).

$$|\psi\rangle = \sum_{j_1,\ldots,j_n=1}^{2} c_{j_i,\ldots,j_n}|j_1,\ldots,j_n\rangle = \sum_{j_1,\ldots,j_n=1}^{2} a_{j_1,\ldots,j_n}|j_1\rangle \otimes \cdots \otimes |j_n\rangle$$





DMRG sweep



$$a_{j_1,\ldots,j_n} = \sum_{\alpha,\ldots,\omega=1}^{D} A^{(1)}_{\alpha;j_1} A^{(2)}_{\beta,\gamma;j_2} \ldots A^{(n)}_{\omega,j_n} = A^{(1)}_{j_1} A^{(2)}_{j_2} \ldots A^{(n)}_{j_n}$$

Borrowed from David Lin's Slides

Quantum annealing

Quick summary
Non-zero temperature and non-adiabatic dynamics.
Solution is obtained probabilistically.

We explore this algorithm on the D-Wave 2000Q.

- 2048 qubits maps to approx. 64 fully connected qubits.
- approx. 10-bits of precision on biases and couplers.

Software suite

`dwave_sapi2.remote`

- connect to DWave

`dwave_sapi2.embedding`

- map *n*-dim Ising model to 3 dimensions.

`dwave_sapi2.util`

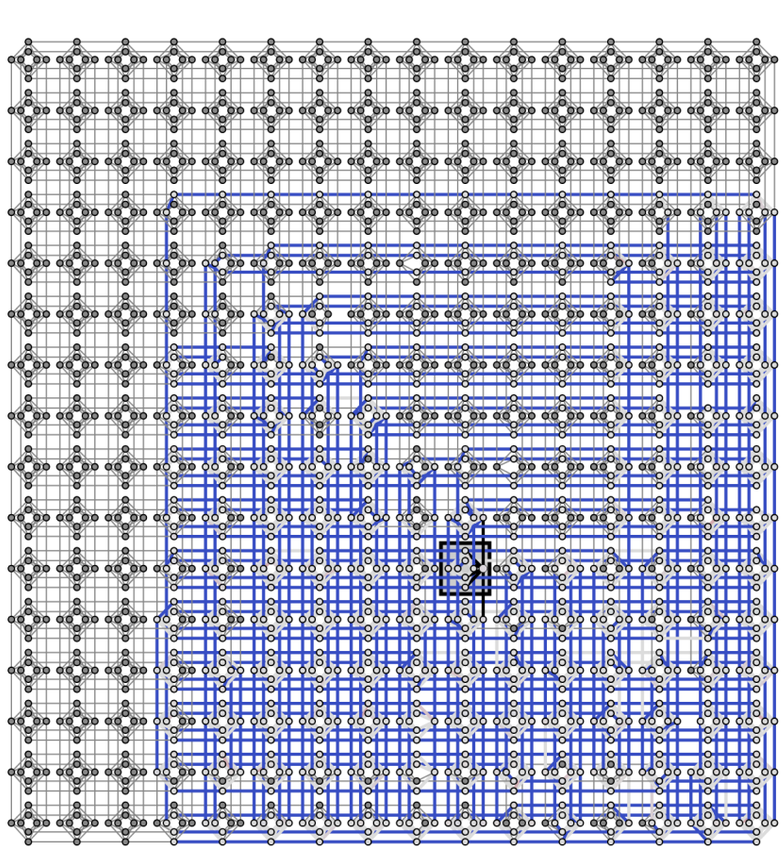
- map QUBO to Ising

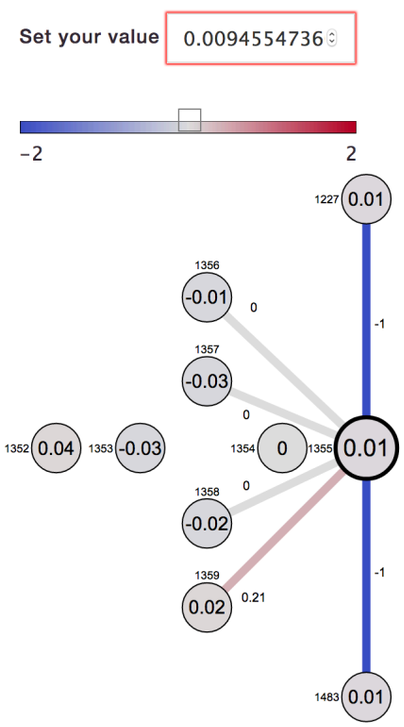
`dwave_sapi2.core`

- calls DWave solver.

`dwave_qbsolv`

- heuristic solver for large problems.





Example: linear least squares

Very simple toy problem to check algorithm.

Generate "data" for the following linear system:

$$D(x) = 8 + 4x + 7x^2$$
$$x \in [0, 1, 2, \ldots, 47, 48, 49]$$

$$\mathrm{Corr}[D] = \begin{pmatrix} 1 & 0.9 & 0.9^2 & \\ 0.9 & 1 & 0.9 & \ldots \\ 0.9^2 & 0.9 & 1 & \\ & \vdots & & \ddots \end{pmatrix}$$

Toeplitz: simulates correlation in time-series

Goal: Solve for the coefficients given data generated above

Fit function

$$f(x) = A_0 + A_1 x + A_2 x^2$$

4-qubit representation of coefficient

$$A_i = \psi_{1,i} + 2\psi_{2,i} + 4\psi_{3,i} + 8\psi_{4,i}$$

Allows for integer values between 0 and 15

3 parameters * 4 spins = 12 total spins or 4096 total solutions

Construct QUBO Hamiltonian and iterate over all solutions (solve for energy and sort).

Result $E_0 = -1.602$ $\Psi = (0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 0)$

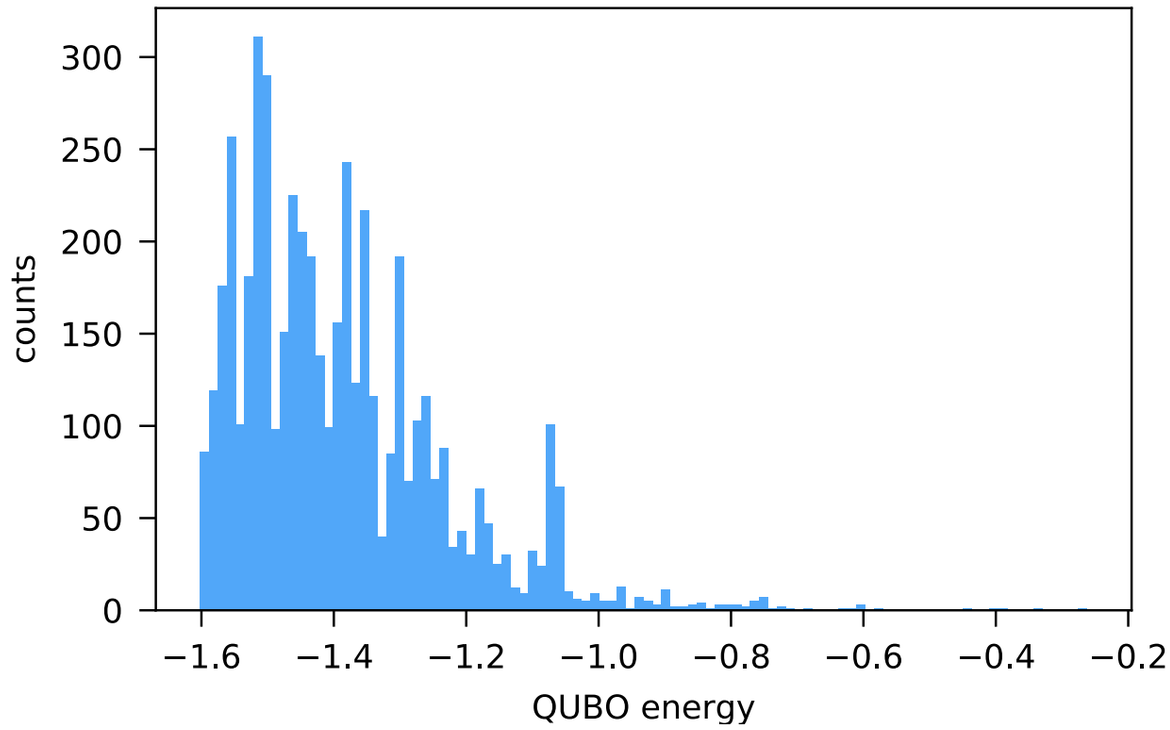
$= (8\ 4\ 7)$

The mapping is correct!

Example: linear least squares

D-Wave 2000Q solution

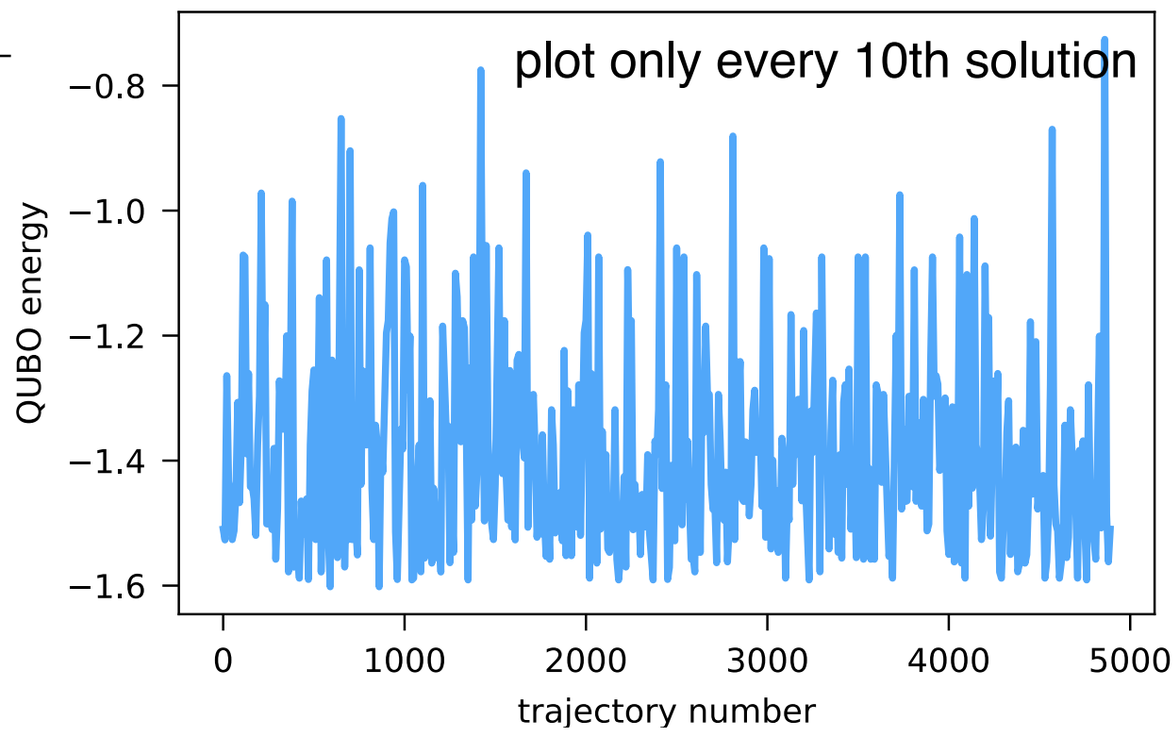
n	brute force E_n	2000Q E_n	count	% hit
0	-1.601898	-1.601898	11	0.22%
1	-1.590899	-1.590899	42	0.86%
2	-1.590882	-1.590882	11	0.22%
3	-1.590253	-1.590253	12	0.24%
4	-1.590176	-1.590176	10	0.20%
5	-1.588166	-1.588166	10	0.20%
6	-1.588107	-1.588107	59	1.20%
7	-1.578343	-1.578343	8	0.16%
8	-1.578264	-1.578264	4	0.08%
9	-1.578132	-1.578132	11	0.22%



- 4900 samples.
- Trajectory suggests stochastic nature of QA.

- Measures correct ground-state energy.
- First energy missed is at $n = 54$.
  - E_{54} = -1.523

Success!



Example: system of 4 linear equations

Conjugate-gradient solution

$$x = \begin{pmatrix} -0.2004 & -0.1075 & 0.1117 & 0.2187 \end{pmatrix}$$

Test a larger problem with inexact solution.

Mapping to QUBO Hamiltonian

CG yields exact solution (the residual is at machine precision)

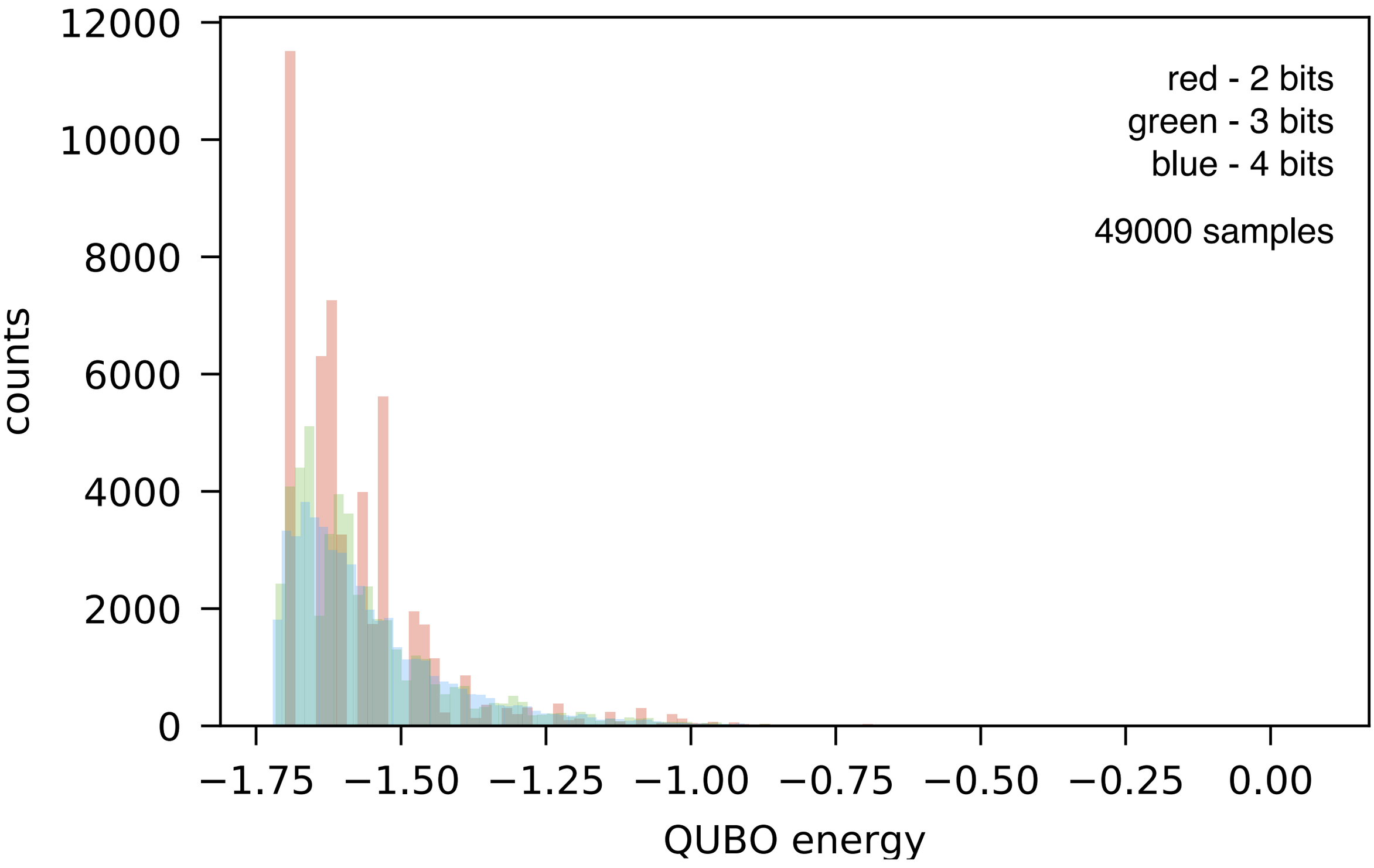
- we can reverse solve for the exact QUBO solution
- Search for solutions between $x_i \in [-0.25, 0.25]$

n-bit	E_0	residual	QA hit rate	range
2	-1.70012	0.3347	23.494%	1E-4
3	-1.71627	0.1842	1.149%	3E-5
4	-1.72149	0.0955	0.094%	6E-6
5	-1.72308	0.0346	—	1E-6
8	-1.72334	0.0058	—	2E-8
16	-1.72334	0.0003	—	3E-13

half-precision
~1E-5

- QUBO matrix with up to half-precision gets correct ground-state
- Residual of algorithm scales exponentially with precision of search.

Example: system of 4 linear equations



Ending remarks

Conclusions

- Present theoretical framework for mapping spin-chain to decimal values.
- System of linear and polynomial equations may be mapped to Ising model.
- Initial study demonstrates validity of algorithm on small test problems.

Future studies

- Quantum annealing scaling as percentage of solutions in the ground state.
- AQC scaling by studying the scale of adiabaticity
- Types of scaling studies
  - varying condition number at fixed size
  - varying size at fixed condition number

Thank you