Adiabatic quantum computing II

Alternative to the standard "gate model" of quantum computing

Adiabatic quantum computation (AQC)

- Engineer a Hamiltonian that changes in time so that $\mathcal{H}(t=0)$ has a known product state as its ground state, e.g.

$$\boxed{|E_0(0)\rangle = |+\rangle|+\rangle\cdots|+\rangle}$$

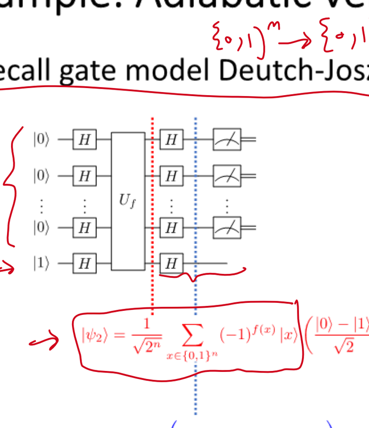
- And for t=t_f, $\mathcal{H}(t_f)$ has a ground state that encodes the solution to the problem you want to solve. E.g. for adiabatic Deutch-Josza:

$$\boxed{|E_0(t_f)\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle}$$

- So that performing a measurement (in a particular basis) will give the desired answer to the problem.

Example: Adiabatic version of Deutch-Josza algorithm

- Recall gate model Deutch-Josza: $\{0,1\}^n \to \{0,1\}$



$$\boxed{|\psi_2\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle}\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

$$|\psi_3\rangle = \sum_{z\in\{0,1\}^n}\left(\frac{1}{2^n}\sum_{x\in\{0,1\}}(-1)^{f(x)+z\cdot x}\right)|z\rangle|1\rangle$$

$$P(z=0) = \left|\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}\right|^2 = \begin{cases}1 & \text{if } f(x) \text{ CONST.}\\ 0 & \text{IF } f(x) \text{ BALANCED.}\end{cases}$$

- Adiabatic version: Want to find Hamiltonian whose ground state is

$$|E_0(t=t_f)\rangle = \boxed{|E_0(1)\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle}$$

a) Starting Hamiltonian:

$$\mathcal{H}(t=0) = \boxed{\mathcal{H}(0) = \omega\sum_{i=0}^{n-1}\left(\frac{I-X_i}{2}\right)}$$

$|E_0(0)\rangle = |+\rangle|+\rangle\cdots|+\rangle$

$\mathcal{H}(0)|E_0(0)\rangle = 0\cdot|E_0(0)\rangle = 0$

$|E_1(0)\rangle = |-\rangle|+\rangle|+\rangle\cdots|+\rangle$

$\mathcal{H}(0)|E_1(0)\rangle = \omega|E_1(0)\rangle,\ E_1(0)=\omega$

$|E_2(0)\rangle = |+\rangle|-\rangle|+\rangle|+\rangle\cdots|+\rangle,\ E_2(0)=\omega$

$|E_3(0)\rangle = |+\rangle|+\rangle|-\rangle|+\rangle\cdots|+\rangle,\ E_3(0)=\omega$

b) We have $\boxed{|E_0(1)\rangle = U|E_0(0)\rangle}$ for $\boxed{U = \sum_{z\in\{0,1\}^n}(-1)^{f(z)}|z\rangle\langle z|:}$

$U|E_0(0)\rangle = \sum_z(-1)^{f(z)}|z\rangle\langle z|\left(\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle\right) = \sum_{x,z}\frac{(-1)^{f(z)}}{\sqrt{2^n}}|z\rangle\delta_{z,x} = \sum_{x\in\{0,1\}^n}\frac{(-1)^{f(x)}}{\sqrt{2^n}}|x\rangle = |E_0(1)\rangle$ //

b) The Hamiltonian $\boxed{\mathcal{H}(1) = U\mathcal{H}(0)U^\dagger}$ has $|E_0(1)\rangle$ as its ground state:

$\mathcal{H}(1)|E_0(1)\rangle = (U\mathcal{H}(0)U^\dagger)U|E_0(0)\rangle = U\mathcal{H}(0)|E_0(0)\rangle = U\cdot 0\cdot|E_0(0)\rangle = 0\ U|E_0(0)\rangle$

MORE GENERALLY, EIGENVECTORS OF $(U\mathcal{H}(0)U^\dagger)$ ARE $U|E_m(0)\rangle$! AND EIGENVALUES ARE E_m (SAME!)

$(U\mathcal{H}(0)U^\dagger)U|E_m(0)\rangle = U\mathcal{H}(0)|E_m(0)\rangle = U E_m(0)|E_m(0)\rangle = E_m(0)\,U|E_m(0)\rangle$

$\Rightarrow$ EIGENVALUES OF $U\mathcal{H}(0)U^\dagger$ ARE ALSO $E_m(0)$!

b) The interpolation between $\mathcal{H}(0)$ and $\mathcal{H}(1)$ can be achieved by:

$$\boxed{\mathcal{H}(s) = \tilde{U}(s)\mathcal{H}(0)\tilde{U}^\dagger(s)},\ \text{where}\ \boxed{\tilde{U}(s) = e^{i\frac{\pi}{2}s(t)U}}$$

$s(t)$ IS SCHEDULE FNC. $s(0)=0,\ s(1)=1$

BECAUSE $U^2 = I \Rightarrow \tilde{U}(s) = \cos\left[\frac{\pi}{2}s(t)\right]I + i\sin\left[\frac{\pi}{2}s(t)\right]U$

$\Rightarrow \tilde{U}(s(0)) = \tilde{U}(0) = I$ AND $\tilde{U}(s(t_f)) = \tilde{U}(1) = iU$

AT $t=t_f$: $\Rightarrow \mathcal{H}(1) = \tilde{U}(1)\mathcal{H}(0)\tilde{U}^\dagger(1) = iU\mathcal{H}(0)(-i)U^\dagger = U\mathcal{H}(0)U^\dagger = \mathcal{H}(1)$!

c) The unitary transf. preserves the energy levels, so $E_1(s)-E_0(s)=\omega$ Hence runtime is set by:

t_f IS RUNTIME!

$$t_f \gg \mathrm{Max}_{s\in[0,1]}\frac{\hbar|\langle E_i(s)|\,\partial\mathcal{H}(s)/\partial s\,|E_0(s)\rangle|}{|E_i(s)-E_0(s)|^2} \sim \frac{\omega}{\omega^2} = \frac{1}{\omega}$$

$\Rightarrow$ INDEPENDENT OF n! ADIABATIC RUNTIME IS $O(1)$, JUST LIKE THE GATE MODEL VERSION!

AQC = gate model up to polynomial overhead!

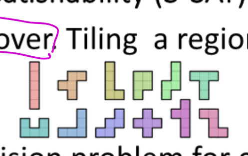
- One can construct a n-qubit circuit that efficiently simulates the AQC time evolution of n qubits with the number of gates scaling as $t_f^2\ \mathrm{poly}(n)$.
  ↳ POLYNOMIAL IN n
- It is also possible to explicitly construct a final Hamiltonian $\mathcal{H}(1)$ whose ground state is , ~~the result of L one~~ and two-qubit gates (circuit with depth L). It was proven that ~~poly(L)~~
  THE RESULT OF L ONE AND TWO $|E_0(1)\rangle = U_L\cdots U_3U_2U_1|00\ldots0\rangle$, TAKES $t_f\sim\mathrm{POLY}(L)$ TO EVOLVE ADIABATICALLY INTO $|E_0(1)\rangle$!
- Which one is better? Depends on the problem. AQC seems really good for optimization problems such as travelling salesman.
- Gate model and AQC are not the only "models" for QC. There are a total of 5 different models:

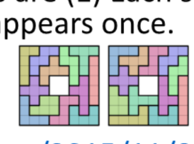
3) Measurement-based QC (done by 1-qubit measurements on a highly entangled initial state)
4) Topological QC (done by braiding anyons on a 2d lattice)
5) Quantum Turing machine

NOT PURSUED FOR HARDWARE, BUT VERY IMPORTANT TO FIND QUANTUM COMPLEXITY CLASSES. (HOW FAST ALGORITHMS CAN BE).

Combinatorial problems: Reduction to 3-satisfiability (3-SAT)

- Example combinatorial problem: Exact cover Tiling a region of 60 squares using each of 12 pentominoes only once.



- This problem can be formulated as a decision problem for choices and constraints: The choices are the different locations and orientations for each pentominoe, and the constraints are (1) Each square must be covered exactly once and (2) each pentominoe appears once.
- Answer : 65 different tilings, e.g. 
- See more at https://garethrees.org/2015/11/09/exact-cover /
- Exact cover is one of the so called **NP-complete** problems. There are hundreds of them, e.g. travelling salesman for integer distances. All of them map into each other with polynomial overhead, so if you solve one you solve all! Another example is the 3-satisfiability (3-SAT) problem that we now show can be solved with AQC.

The 3-satisfiability (3-SAT) problem

- 3-SAT: Find the n-bit string $z_{n-1}z_n\ldots z_0$ that satisfies a set of 3-bit constraint clauses. Each clause involves 3 bits i, j, k and the constraint is that one of the 3 bits must have value 1 and the other two must be 0:

$$\boxed{c(i,j,k) = z_i + z_j + z_k - 1 = 0}$$

- An n-bit instance of 3-SAT is a list of triples (i, j, k) indicating which groups of three bits are involved in clauses. The problem is to determine whether there is some assignment of the n-bit values that satisfies all of the clauses.
- Given an assignment of values for $z_{n-1}z_n\ldots z_0$, we can easily check whether the assignment satisfies all of the clauses. But determining whether at least one of the 2^n assignments of $z_{n-1}z_n\ldots z_0$ satisfies all the clauses is in fact an NP-complete problem. Example: $n=5 \Rightarrow z_4z_3z_2z_1z_0$

CLAUSES: $c_a(0,1,2)$, $c_b(1,2,3)$, $c_c(0,3,4)$

z_4	z_3	z_2	z_1	z_0	
1	0	0	1	0	✓
1	0	1	0	0	✓
1					✗
1	0			0	✗
1	1	0	0	1	✗

ONLY TWO!

AQC algorithm for 3-SAT E. Farhi et al, Science **292**, 472 (2001)

READ THIS IF YOU'RE "CORONA-BORED"!

a) Starting Hamiltonian:

$$\mathcal{H}(t=0) = \mathcal{H}(0) = \omega\sum_{i=0}^{n-1}\left(\frac{I-X_i}{2}\right)$$

USUAL ONE, $|E_0(0)\rangle = |+\rangle|+\rangle\cdots|+\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle$

b) Final Hamiltonian:

$$\mathcal{H}(t=t_f) = \mathcal{H}(1) = \sum_{\text{all } C(i,j,k)}\left[\underbrace{\left(\frac{I-Z_i}{2}\right)}_{=z_i}+\underbrace{\left(\frac{I-Z_j}{2}\right)}_{=z_j}+\underbrace{\left(\frac{I-Z_k}{2}\right)}_{=z_k}-1\right]^2$$

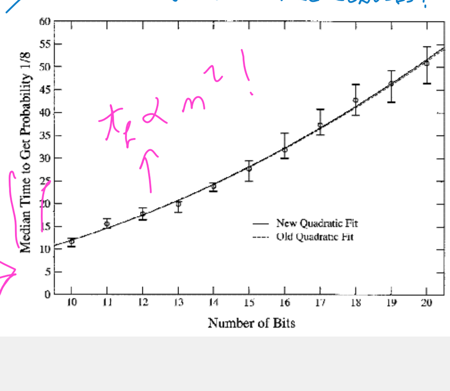
$= \sum_{\text{all } C(i,j,k)}(z_i+z_j+z_k-1)^2$

$E_0(1) = 0$ WHEN $|z_{n-1}z_{n-2}\cdots z_0\rangle$ SATISFIES ALL CLAUSES!

c) Picture shows median t_f required to get the right answer with prob. 1/8 using exact integration of Schröedinger eqn for linear s(t):

$$\boxed{\mathcal{H}(t) = \left(1-\frac{t}{t_f}\right)\mathcal{H}(0) + \frac{t}{t_f}\mathcal{H}(1)}$$

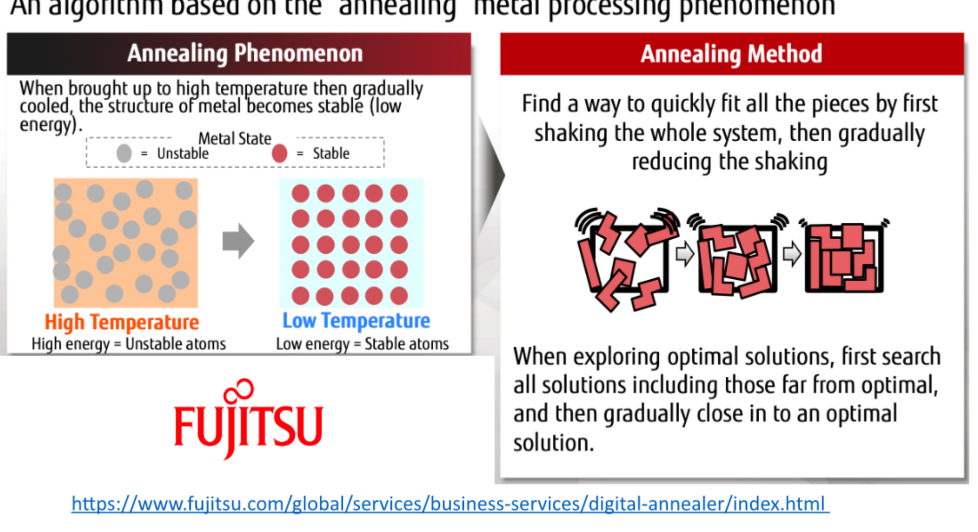
$s(t) = \frac{t}{t_f}$, CAN CHOOSE OTHER ONES!



Annealing method: A heuristic algorithm

= TRIAL AND ERROR.

An algorithm based on the "annealing" metal processing phenomenon

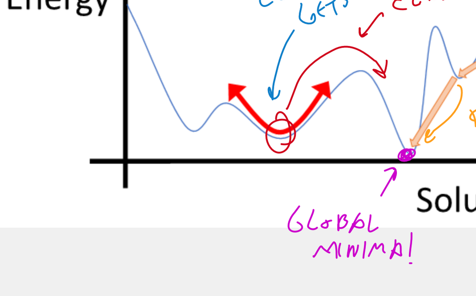


https://www.fujitsu.com/global/services/business-services/digital-annealer/index.html

Quantum annealing (QA) as heuristic AQC

- In Quantum annealing, one performs a **"loose run"** of the 3-SAT AQC for particular choices of s(t) and t_f, without worrying whether the system stays in the ground state or not. One experiments with s(t) and t_f with the goal of optimizing the probability of getting the right answer to the problem. There are no guarantees that QA works in all cases, it's trial and error (heuristics)!
- Why QA may have "quantum advantage" over classical annealing algorithms: Energy landscape metaphor. (CARTOON)

CLASSICAL ANNEALING GETS TRAPPED IN LOCAL MINIMA! CLASSICAL SYSTEM MUST OVERCOME ENERGY BARRIER TO FIND SOLUTION!



QUANTUM MECH ALLOWS YOU TO TUNNEL THROUGH BARRIER!

GLOBAL MINIMA!

3-SAT is a particular case of QUBO

- Let's expand the 3-SAT $\mathcal{H}(1)$:

$$\mathcal{H}(1) = \sum_{\text{all } C(i,j,k)}\left[\left(\frac{I-Z_i}{2}\right)+\left(\frac{I-Z_j}{2}\right)+\left(\frac{I-Z_k}{2}\right)-1\right]^2$$

$= \sum_{\text{all}}\left[z_i^2+z_j^2+z_k^2+1-2(z_i+z_j+z_k)+2(z_iz_j+z_iz_k+z_jz_k)\right]$

$z_i^2 = z_i$ (BECAUSE $z_i\in\{0,1\}$)

$$= N_c - \sum_i B_i z_i + 2\sum_{i<j}C_{ij}z_iz_j$$

N_c = # OF CLAUSES; B_i IS # OF TIMES z_i APPEARS IN ALL CLAUSES; C_{ij} IS # OF TIMES PAIR i,j APPEARS IN ALL CLAUSES

- This is a binary quadratic form. So 3-SAT is a particular case of

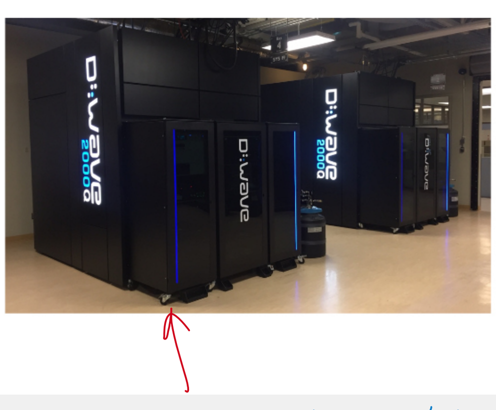
Quadratic Unconstrained Binary Optimization (QUBO)

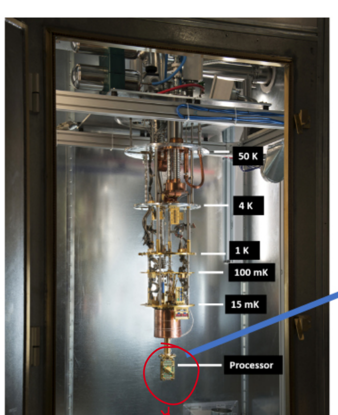
$$\mathcal{H}_{QUBO}(1) = -\sum_i b_i z_i + \sum_{i<j}c_{ij}z_iz_j$$

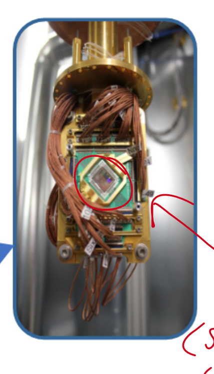
CHOOSE YOUR b_i AND c_{ij} AS YOU WISH.

D-Wave Leap: Cloud access to QA hardware

- D-Wave from Burnaby, BC is the leading developer of QA hardware and software. Next class we will access their devices using cloud-based D-Wave's platform, D-Wave Leap.







2000 COUPLED FLUX QUBITS (SUPERCONDUCTING CIRCUIT TECHNOLOGY).

HOW COME THEY ALREADY HAVE 2000 QUBITS?
→ BY FOCUSING ON A SPECIFIC ALGORITHM (QUBO) MUCH LESS CONTROL IS REQUIRED! AND ALSO, QA IS MORE ROBUST TO NOISE. SO D-WAVE IS ABLE TO SCALE UP FASTER THAN THEIR COMPETITION!