Lecture 7: Heuristic Algorithms

We have now seen the three main classes of quantum algorithm whose performance can be analysed rigorously.

However, all of these algorithms require large-scale ($\Rightarrow$ fault-tolerant) quantum computers to run them on useful size problem instances.

We are only barely entering the NISQ era of Noisy, Intermediate-Scale Quantum computers.

In recent years, a number heuristic q. algorithms have been proposed. Whilst they are heuristic — i.e. their performance is hard to analyse mathematically, and there is far less certainty they outperform classical algorithms — they have the advantage that they are potentially easier to run on smaller-scale q. hardware, and see how they perform in practice.

Adiabatic QC

Thm (Adiabatic Thm)
Let $H(s)$, $s \in [0,1]$ be a smooth function of s s.t. $H(s)$ always has a non-degenerate g.s.

Let $|\phi(s)\rangle$ = g.s. of $H(s)$

$\Delta(s)$ = $\lambda_1(H(s)) - \lambda_0(H(s))$
spectral gap of $H(s)$

$\dot{H}(s) = \left.\frac{dH}{ds}\right|_s$

Taking $s(t) = \frac{t}{T}$, let
$|\psi(s)\rangle$ = solution to time-dep. Schrödinger eq:

$$\frac{\partial |\psi(s)\rangle}{\partial t} = -i H(s) |\psi(s)\rangle$$

If s is varied slowly enough that total evolution time

$$T \geq \frac{1}{\varepsilon}\left[\frac{\|H(0)\|}{\Delta(0)^2} + \frac{\|\dot{H}(1)\|}{\Delta(1)^2} + \int_0^1 ds \left(\frac{\|\dot{H}\|^2}{\Delta^3} + \frac{\|\ddot{H}\|^2}{\Delta^2}\right)\right]$$

then $\| |\phi(s)\rangle - |\psi(s)\rangle \| \leq \varepsilon$.

Thm [Aharonov et al. '04]

Adiabatic QC is polynomially equivalent to circuit model.

(But only theoretically; H required in proof is highly contrived.)

In practice, adiabatic QC is used for optimisation problems, not universal QC.

Advantages:

- Easier to implement in devices ?

Disadvantages:

- No know method of making it fault-tolerant
- Not well-suited to universal QC
- Not clear adiabatic optimization beats classical computation on any problem.

QAOA

Q. Adiabatic / Approximate Optimisation Alg.

Classical constraint satisfaction problems can be cast as minimising the "cost" of violating the constraints.

E.g. satisfying Boolean constraints

$$\begin{cases} a \wedge b = 1 \\ \neg b \wedge c = 1 \end{cases}$$

equivalent to $\min_{a,b,c} C(a,b,c)$ where

$$C(a,b,c) = (\neg a \wedge b + b \wedge \neg c).$$

A "good" solution minimises cost function C = # violated constraints.

Note: $C(a,b,c)$ is just a classical k-local Hamiltonian ($k=2$ in above example).
Minimising C = minimising energy.

Problem (Boolean Constraint optimisation)

Input: Boolean constraints
$\{C_j(b_1^{(j)}, \ldots, b_k^{(j)})\}$, $j = 1 \ldots m$

Output: bit string $\vec{b}$ s.t.
$\sum_j C_j(\vec{b})$ is small

QAOA algorithm

$$U_C(\gamma) := \prod_{j=1}^{m} e^{-i\gamma C_j}$$

$$U_B(\beta) := \prod_{i=1}^{n} e^{-iX^{(i)}}$$

$$|\vec{\gamma}, \vec{\beta}\rangle := \prod_{j=1}^{p} U_B(\beta_j) U_C(\gamma_j) \left(\frac{1}{\sqrt{2^n}} \sum_x |x\rangle\right)$$

$\vec{b}$ = outcome of measuring $|\vec{\gamma}, \vec{\beta}\rangle$ in computational basis.

Want $C(\vec{\gamma}, \vec{\beta}) = \langle \vec{\gamma}, \vec{\beta} | \sum_j C_j | \vec{\gamma}, \vec{\beta} \rangle$ small

→ run classical optimisation over $\vec{\gamma}, \vec{\beta}$, using q. computer to compute $C(\vec{\gamma}, \vec{\beta})$.

Analysis

Can prove

$$\lim_{p\to\infty} \langle \vec{\gamma}, \vec{\beta} | \sum_j C_j | \vec{\gamma}, \vec{\beta} \rangle = \min_{\vec{b}} \sum_j C_j(\vec{b})$$

However, finding optimal solution (and even getting within factor of $1-\varepsilon$ of optimal solution, for ε depending on type of constraints involved) is NP-hard in general.

$\longrightarrow$ $p = O(2^n)$ in worst case.

Advantages:

- Can choose p = whatever circuit depth hardware is capable of.
- Numerical simulations indicate QAOA often works well even for small p = low-depth circuits (NISQ?)

Disadvantages:

- Not known to beat best classical algorithms, even theoretically
- Classical heuristic constraint optimisation algorithms work very well in practice, even on large problems.

When it first came out, QAOA was proven to beat best classical optimisation algorithm for MAXSAT.

Within ~2 months, classical algorithm was found that beats it.

VQE

Variational Q. Eigensolver

Thm. (Variational characterisation of eigvals.)

Hamiltonian H.

$$\lambda_0(H) = \min_{|\psi\rangle} \langle\psi|H|\psi\rangle$$
$$\leq \langle\varphi|H|\varphi\rangle \quad \forall\, |\varphi\rangle$$

VQE algorithm

Local Hamiltonian $H = \sum_j h_j$.

Let $|\psi\rangle = U_C\,|00\ldots0\rangle$
for some class of q. circuits U_C.

Run classical optimisation over $\vec{\gamma}, \vec{\beta}$, using q. computer to compute energy (a.k.a. cost function):

$$\langle\psi|H|\psi\rangle = \langle 0\ldots0|\, U_C^\dagger \left(\sum_j h_j\right) U_C\, |0\ldots0\rangle.$$

by measuring h_j on $U_C\,|0\ldots0\rangle$.

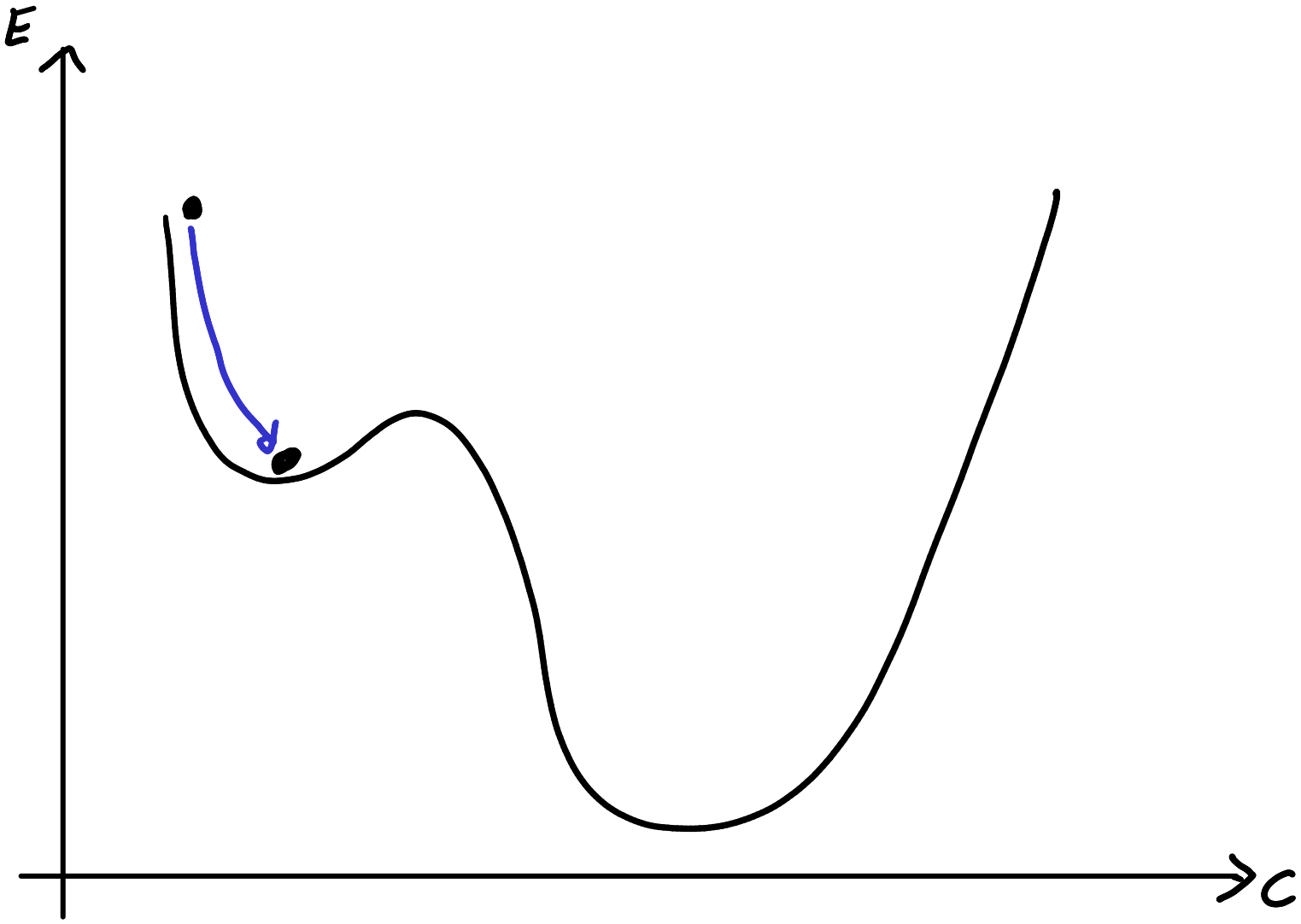
Note: QAOA can be seen as a special case of this

Advantages:

- Can let U_c be whatever class of circuits you can perform on the device (NISQ?)
  $\longrightarrow$ can always get <u>some</u> result out.
- Best current classical algorithms for finding g.s. energies are also variational, but over classes of q. states that can be described classically (DFT, Q. Monte-Carlo, Tensor networks, etc.)

Disadvantages:

- Hard to know how good/bad variational result is.
  (see fig. for illustration)
- Classical optimisation step is hard, both theoretically (QMA-hard!) and in practice.
- Hard to know if it will beat existing classical methods, except by running it and seeing.


E
C