Quantum Approximate Optimization Algorithm

assorted stuff

Edward Farhi

Google

MIT

Jeffrey Goldstone Sam Gutmann

Combinatorial Optimization

n bits $\qquad$ m clauses

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z) \qquad z_1 \ldots z_n = z$$

$$C_\alpha(z) = \begin{cases} 1 & \text{satisfies} \\ 0 & \text{does not} \end{cases}$$

$$C_{max} = \max_z C(z)$$

Want $\boxed{\dfrac{C(z)}{C_{max}}}$ big.

Quantum Algorithm Ingredients:

$$\underline{U(C,\gamma)} = e^{-i\gamma C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha}$$

$$B = \sum_{j=1}^{n} X_j \qquad X_j = \sigma_j^x$$

$$\underline{U(B,\beta)} = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta X_j}$$

$$\underline{|s\rangle} = \frac{1}{\sqrt{2^n}} \sum_z |z\rangle$$

$$|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

All easy to construct!

For any integer $p \geq 1$ $\underline{\gamma_1 \ldots \gamma_p = \vec{\gamma}}$ $\underline{\beta_1 \ldots \beta_p = \vec{\beta}}$

$$\underline{|\vec{\gamma}, \vec{\beta}\rangle} = U(B, \beta_p)\, U(C, \gamma_p) \cdots U(B, \beta_1)\, U(C, \gamma_1)\, |s\rangle$$

Required Circuit Depth at most $mp + p$.

$$F_p(\vec{\gamma}, \vec{\beta}) = \langle \vec{\gamma}, \vec{\beta} | C | \vec{\gamma}, \vec{\beta} \rangle$$

$$M_p = \max_{\vec{\gamma}, \vec{\beta}} F_p(\vec{\gamma}, \vec{\beta})$$

$$\boxed{M_p \geq M_{p-1}}$$

Can Show $\boxed{\lim_{p \to \infty} M_p = C_{max}}$

Quantum Algorithm with Angle Search

Fix p. Start with angles $(\vec{\gamma}, \vec{\beta})$

Use the Quantum Computer to make

$$|\vec{\gamma}, \vec{\beta}\rangle$$

Measure to get a string z and $C(z)$.

Repeat with same angles to get a good estimate of $F_p(\vec{\gamma}, \vec{\beta})$

Repeat with new angles to get near

$$M_p = \max_{\vec{\gamma}, \vec{\beta}} F_p(\vec{\gamma}, \vec{\beta})$$

For p fixed we can classically preprocess and determine the best angles in advance.

Example: Max Cut on 3-regular graphs

$$C = \sum_{\langle jk \rangle} C_{\langle jk \rangle}$$

$$C_{\langle jk \rangle} = \frac{1}{2}\left(-Z_j Z_k + 1\right) \qquad Z_j = \pm 1$$

$p=1$ Look at contribution from edge $\langle jk \rangle$

$$\langle s | e^{i\gamma C} e^{i\beta B} Z_j Z_k e^{-i\beta B} e^{-i\gamma C} | s \rangle$$

$$|s\rangle = |+\rangle_1 |+\rangle_2 \cdots |+\rangle_n$$

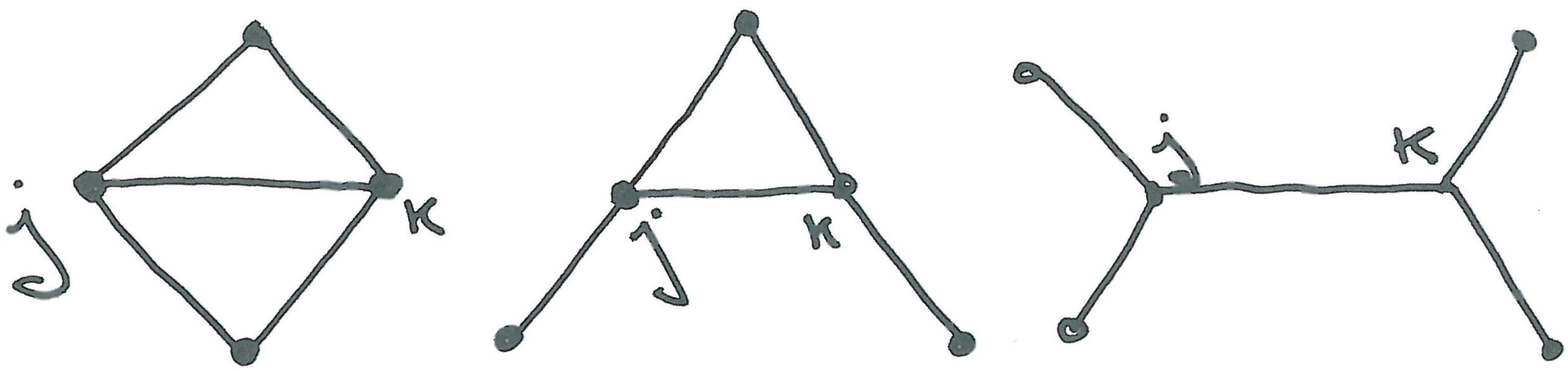
$$e^{i\beta B} Z_j Z_k e^{-i\beta B} = e^{i\beta(X_j+X_k)} Z_j Z_k e^{-i\beta(X_j+X_k)}$$

$$= [\cos 2\beta Z_j + \sin 2\beta Y_j][\cos 2\beta Z_k + \sin 2\beta Y_k]$$

only bits j k involved

Conjugate with $e^{i\gamma C}$

only bits connected to j, k involved



3 possible subgraphs

Each subgraph type gives a function of γ, β which does not depend on n or m.

These can be evaluated on a classical computer looking at a 4, 5 or 6 qubit system. Then

$F_1(\gamma, \beta)$

can be evaluated on a classical computer and optimal angles chosen.

At $p=1$, the QAOA will produce a cut that is at least .6924 times the optimal cut. For ALL instances of 3-regular Max Cut!

Max E3LIN2

n variables, m equations each with 3 variables $(z_i + z_j + z_k) \bmod 2 = \begin{cases} 1 \\ 0 \end{cases}$ $z_i = 0, 1$

Instance is specified by a collection of triples and a 0 or 1 for each triple

Task Find a string that maximizes the number of satisfied equations. NP hard to find the optimal solution.

Try to find a "good" solution.

Approximate Optimization

Algorithm: Guess a Random String
Achieves $\frac{1}{2}$

Limit: $\left(\frac{1}{2}+\varepsilon\right)$ would imply P=NP 2001

Bounded Occurrence: Every variable is in no more than D equations

Algorithm: $\left(\frac{1}{2}+\frac{\text{const}}{D}\right)$ 2000

Quantum Algorithm: $\left(\frac{1}{2}+\frac{\text{const}}{D^{3/4}}\right)$ 2014

Classical Algorithm: $\left(\frac{1}{2}+\frac{\text{const}}{D^{1/2}}\right)$ 2015

Boaz Barak, Ankur Moitra,
Ryan O'Donnell, Prasad Raghavendra,
Oded Regev, David Steurer,
Luca Trevisan,
Aravindan Vijayaraghavan,
David Witmer, John Wright,
Johan Håstad

Quantum Approximate Optimization Algorithm

E.F., Jeffrey Goldstone, Sam Gutmann

Better Analysis: $\left(\frac{1}{2} + \frac{1}{101\, D^{1/2} \ln D}\right)$

2015

Typical: $\left(\frac{1}{2} + \frac{1}{2\sqrt{3e}\, D^{1/2}}\right)$

Can we get rid of the $\ln D$???

If $\exists$ algorithm $\left(\frac{1}{2} + \frac{\text{const}}{D^{1/2}}\right)$

for a sufficiently large constant

then P=NP.

Sherrington - Kirkpatrick Model

n bits $z_a = \pm 1$

$$\left[C_J(z) = \frac{1}{\sqrt{n}} \sum_{a<b} J_{ab} z_a z_b \right]$$

mean $J_{ab} = 0$ var $J_{ab} = 1$

$$\left[\lim_{n \to \infty} E_J \max_z \frac{C_J(z)}{n} = .763\ldots \right]$$

Parisi !

Stat Mech Approach

at low temperature T evaluate

$$\frac{T}{n} E_J \left[\log \left(\sum_z \exp\left(-\frac{C_J(z)}{T} \right) \right) \right]$$

Hard because of E_J of $\log(\ )$

We avoid this!

P=1 QAOA

$$\langle s| e^{i\gamma C_J} e^{i\beta B} \frac{C_J}{n} e^{-i\beta B} e^{-i\gamma C_J} |s\rangle$$

$$= \frac{1}{2^n} \sum_{z^1} \sum_{z^2} \sum_{z^3} e^{i\gamma C_J(z^1)} \langle z^1| e^{i\beta B} |z^2\rangle \cdot$$

$$\frac{1}{n} C_J(z^2) \langle z^2| e^{-i\beta B} |z^3\rangle e^{-i\gamma C_J(z^3)}$$

---

again $\quad C_J(z) = \frac{1}{\sqrt{n}} \sum_{a<b} J_{ab} z_a z_b$

Now for one Gaussian J

$$\left[\begin{aligned} E_J\, e^{i\phi J} &= e^{-\phi^2/2} \\ E_J\, e^{i\phi J} J &= i\phi\, e^{-\phi^2/2} \end{aligned}\right]$$

Upstairs $\gamma \sum_{a<b} J_{ab}(Z_a^1 Z_b^1 - Z_a^3 Z_b^3)$

Downstairs $\gamma \sum_{c<d} J_{cd} Z_c^2 Z_d^2$

Turn Crank

$$\left[E_J \langle \gamma, \beta | \frac{C_J}{n} | \gamma, \beta \rangle_J \text{ as } n \to \infty \right.$$

$$\left. = \underline{2 \sin 2\beta \cos 2\beta \ \gamma e^{-2\gamma^2}} \right]$$

Optimum at $\beta = \pi/8$ $\gamma = 1/2$

get $\boxed{\frac{1}{2\sqrt{e}} \approx .303}$

For any fixed P we have an iterative procedure to evaluate

$$\left[\lim_{n\to\infty} E_J \, {}_J\langle \vec{\gamma}, \vec{\beta} | \frac{C_J}{n} | \vec{\gamma}, \vec{\beta} \rangle_J \right]$$

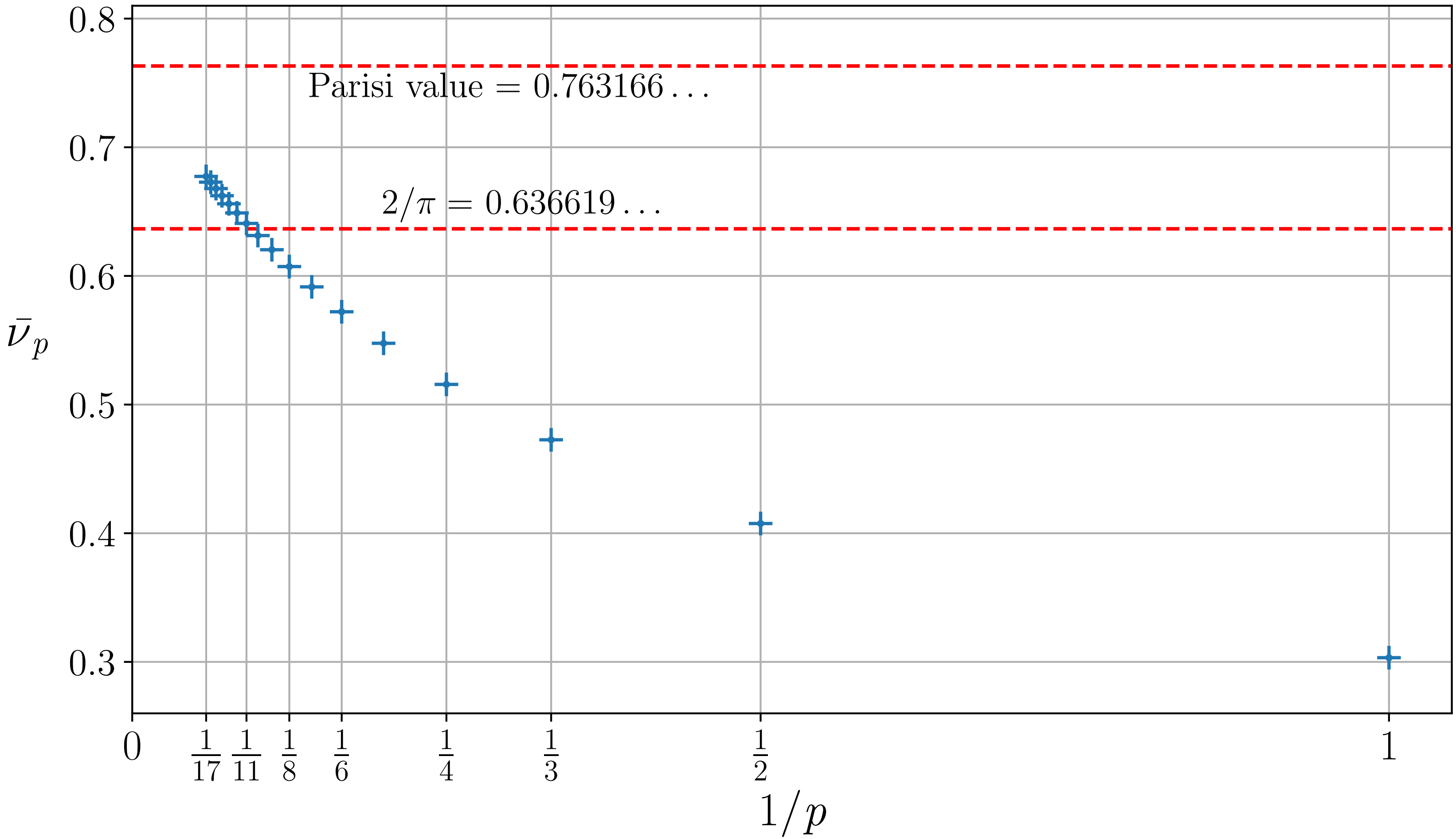
which takes $\sim 4^P$ steps

E. F.
S. Gutmann
J. Goldstone
L. Zhou

Again P fixed $n \to \infty$.

Call this quantity

$$V_P^{SK}(\vec{\gamma}, \vec{\beta})$$



Parisi value = 0.763166...
2/π = 0.636619...
$\bar{\nu}_p$
0.8
0.7
0.6
0.5
0.4
0.3
0
1/17
1/11
1/8
1/6
1/4
1/3
1/2
1
1/p

Also show for any $\vec{\gamma}, \vec{\beta}$ at any depth p

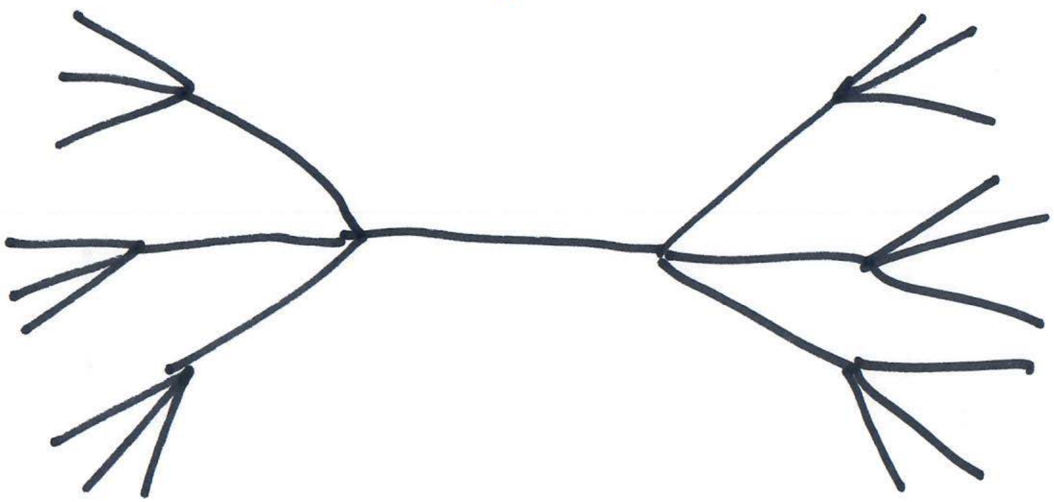
$$\lim_{n\to\infty} E_J\left[\langle \vec{\gamma}, \vec{\beta} | \left(\frac{C_J}{n}\right)^2 | \vec{\gamma}, \vec{\beta} \rangle\right]$$
$$= \left\{\lim_{n\to\infty} E_J\left[\langle \vec{\gamma}, \vec{\beta} | \frac{C_J}{n} | \vec{\gamma}, \vec{\beta} \rangle\right]\right\}^2$$

which proves Landscape Independence for the SK Model.

Can also show that for $n \to \infty$, for typical instances all measured strings z have

$$\left\{ \frac{C(z)}{n} \text{ close to } \langle \vec{\gamma}, \vec{\beta} | \frac{C}{n} | \vec{\gamma}, \vec{\beta} \rangle \right\}.$$

Consider Max-Cut on large-girth D-regular graphs. Apply QAOA
For fixed p see only a tree neighborhood



One subgraph

$$\frac{1}{|E|}\langle\vec{\gamma},\vec{\beta}|\,C_{MC}\,|\vec{\gamma},\vec{\beta}\rangle$$

$$= \frac{1}{2} + \frac{\nu_p(D,\vec{\gamma},\vec{\beta})}{\sqrt{D}}$$

Now

$$\boxed{\lim_{D\to\infty} \nu_p(D,\vec{\gamma},\vec{\beta}) = V_p^{SK}(\vec{\gamma},\vec{\beta})}$$

one subgraph = average over instances !

J. Basso, E.F., K. Marwaha,

B. Villalonga, Leo Zhou

Outstanding Paper Prize

TQC 2022

The QAOA exhibits Quantum Supremacy!!

with Aram Harrow

$$\tilde{H} = e^{-i\frac{\pi}{4}\sigma_x}$$

C = a sum of two bit clauses

$$|\psi\rangle = \tilde{H}^{\otimes n} e^{-i\gamma C} |s\rangle \qquad p=1$$

$$q(z) = |\langle z|\psi\rangle|^2$$

Suppose You have a classical algorithm that outputs strings with probability $p(z)$ and

$$|q(z) - p(z)| < q(z)/10$$

Then Polynomial Hierarchy Collapses!

That was worst case.

For average or typical use SK cost function.

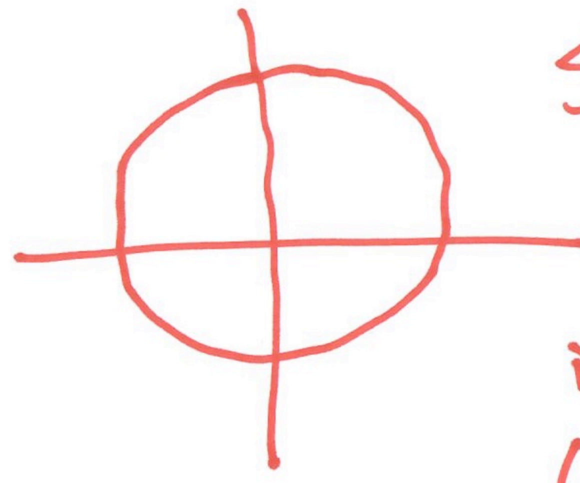
Just showed concentration over measurement outcomes. Bad for supremacy arguments.

$C_J = \frac{1}{\sqrt{n}} \sum_{a<b} J_{ab} z_a z_b$, $e^{i\gamma C_J}$, γ of order 1

Drop $\frac{1}{\sqrt{n}}$ $\quad \bar{C}_J = \sum_{a<b} J_{ab} z_a z_b$, $e^{i\bar{\gamma} \bar{C}_J}$, $\bar{\gamma}$ of order 1

Look at one term

$$e^{i\bar{\gamma} J_{12} z_1 z_2}$$



Spread out over the circle in the complex plane

Typical Case Supremacy requires
[$\bar{\gamma}$ of order unity]

States look random.

Cost function concentrates at 0

For optimization need
[$\bar{\gamma}$ of order $\frac{1}{\sqrt{n}}$]

QAOA can work as an optimizer
and have Supremacy !

The QAOA on a bounded degree graph at low depth is a local algorithm

General arguments that apply to any local algorithm can be applied (with care).

Overlap Gap?

What does local mean??

Distant Qubits

On the instance graph qubit i is at least $2p$ away from qubit j.

$$|\vec{\alpha}, \vec{\beta}\rangle = U_p |s\rangle$$

[$|s\rangle$ is a product state]

U_p is the depth p QAOA circuit unitary

$$\langle s| U_p^\dagger \hat{O}_i \hat{O}_j U_p |s\rangle = \langle s| U_p^\dagger \hat{O}_i U_p |s\rangle \langle s| U_p^\dagger \hat{O}_j U_p |s\rangle$$

$\hat{O}_i$ is any operator acting on qubit i

Measurements on i and j are independant.

Far from an Edge

Edge ij. Call Far those bits at least p away from ij.

Let U_p be the QAOA unitary.
Modify the cost function on edge ij.
New unitary U_p^M.

Then $$U_p^{M+} \hat{O}_{Far} U_p^M = U_p^+ \hat{O}_{Far} U_p$$

Measurements in Far are unaffected by the change.

The QAOA needs to see the whole graph !

Locality + other Stuff can be used to give upper bounds on performance.

Example Maximum Independent Set on a d-regular random graph

Each vertex "sees" roughly d^p other vertices.
Trouble if this is less than n.

We give an upper bound for

$$\boxed{p < \text{const} \cdot \log n / \log d}$$

Put numbers in $n = 10^6$, $d = 3$ get $p < 7$

On dense graphs there are failure bounds if $\boxed{p < \text{constant} \cdot \log \log n}$

No indications of failure if

$\boxed{p > \text{const} \cdot \log n}$ for a big enough constant

Anshu, Metger

The QAOA gets stuck starting from a good classical string!

M. Cain, E.F., S. Gutmann, D. Ranard, E. Tang

Usual QAOA starts in

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum_z |z\rangle$$ superposition of all inputs

$$= |+\rangle_1 |+\rangle_2 \cdots |+\rangle_n$$ product state

$$|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

Run a classical algorithm and get a "good" string $w = w_1 \ldots w_n$

Simulated Annealing

Goemans-Williamson

or whatever.

$$U(\vec{\gamma}, \vec{\beta}) = e^{-i\beta_p X} e^{-i\gamma_p C} \ldots e^{-i\beta_1 X} e^{-i\gamma_1 C}$$

act on $|w\rangle$

$$U(\vec{\gamma}, \vec{\beta})|w\rangle$$

Now look for good parameters

Dramatic Failure

! At low bit number simulation we can not !
! find any improvement beyond $C(w)$. !

For example. 16 bit 3-regular graph

24 edges

average cut value 12

Start mo $|w\rangle$'s where $C(w)=15$

Look at $P = 1, 2, 3 \ldots$

Can not find parameters that improve cost beyond 15 (maybe by .2)

Slight Intuition

$$C_{MC} = \sum_{\langle ij \rangle} \frac{1}{2}(1 - z_i z_j) = \sum_{\langle ij \rangle} C_{ij}$$

$\langle w | U^\dagger C_{ij} U | w \rangle$

$U = 1$ 1 if $w_i \neq w_j$

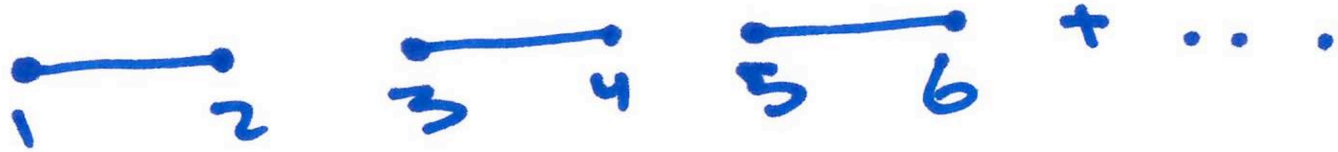
     0 if $w_i = w_j$

$U \neq 1$ $1 - \varepsilon$ if $w_i \neq w_j$

     δ if $w_i = w_j$

Suppose $\varepsilon = \delta$. In a good string there are more satisfied edges than unsatisfied. More edges get worse than those that improve. Net loss

Decoupled graph



For any QAOA unitary U we can
prove that $\varepsilon = \delta$ for each •—•
At any depth starting in a good string
fails. However $p=1$ QAOA starting
in $|s\rangle$ works perfectly!
(Even with $\pm Z_i Z_j$ for each edge!)

We also have beautiful thermal
arguments for more general
cases!

The fact that you cannot warm start the QAOA from a single computational basis state is counterintuitive and suggests that there is something special about starting from

$$\boxed{|s\rangle = \frac{1}{\sqrt{2^n}} \sum_z |z\rangle}$$

from which progress is always made

* Easy to implement all purpose optimizer
* Worst case performance guarantees
* Can analyze random instances: finite and infinite size
* Supremacy: worst case and typical
* Performance improves with depth: Theory and numerics
* Possible that at attainable depth on a near term or scalable fault tolerant computer the QAOA will outperform classical algorithms for some task!
* Only know if you try!