The Quantum Approximate Optimization Algorithm and the Sherrington-Kirkpatrick Model at Infinite Size

Leo Zhou (Harvard University)

with Edward Farhi, Jeffrey Goldstone, and Sam Gutmann

QIP – Feb 1, 2021

Combinatorial Optimization Problems

Cost function

$$C(\boldsymbol{z}) = \sum_{\alpha} C_{\alpha}(\boldsymbol{z})$$

$$\boldsymbol{z} = (z_1, \ldots z_n) \in \{\pm 1\}^n$$

Want $\boldsymbol{z}^*$ so $C(\boldsymbol{z}^*)$ is maximized

Combinatorial Optimization Problems

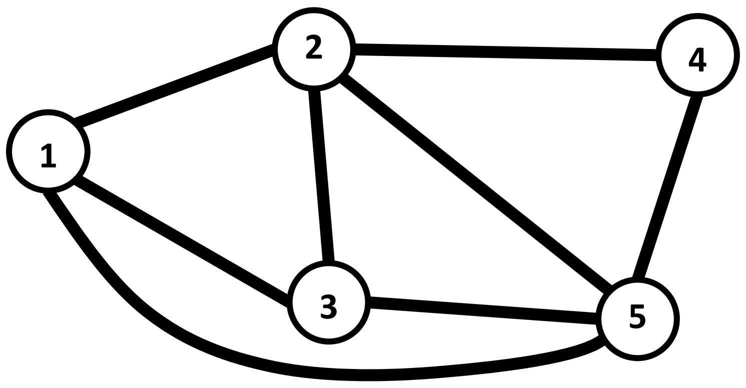
Cost function

$$C(\boldsymbol{z}) = \sum_{\alpha} C_\alpha(\boldsymbol{z})$$

$$\boldsymbol{z} = (z_1, \ldots z_n) \in \{\pm 1\}^n$$

Want z^* **so** $C(z^*)$ **is maximized**

MaxCut



Goal: find a **bipartition** of vertices that cut the maximum # edges

Combinatorial Optimization Problems

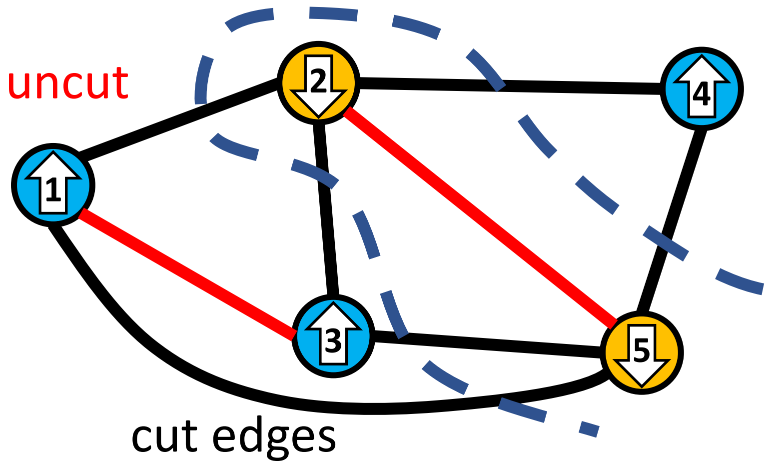
Cost function

$$C(\boldsymbol{z}) = \sum_{\alpha} C_{\alpha}(\boldsymbol{z})$$

$$\boldsymbol{z} = (z_1, \ldots z_n) \in \{\pm 1\}^n$$

Want z^* **so** $C(z^*)$ **is maximized**

MaxCut



Goal: find a **bipartition** of vertices that cut the maximum # edges

Combinatorial Optimization Problems

Cost function

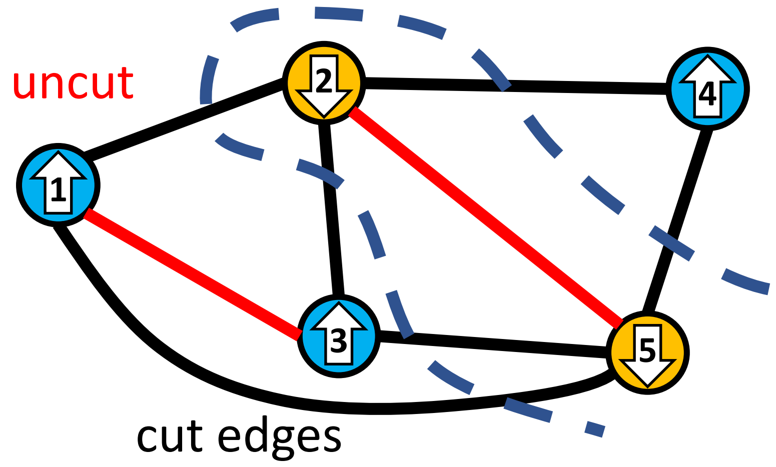
$$C(\boldsymbol{z}) = \sum_{\alpha} C_\alpha(\boldsymbol{z})$$

$$\boldsymbol{z} = (z_1, \ldots z_n) \in \{\pm 1\}^n$$

Want z^* **so** $C(z^*)$ **is maximized**

MaxCut

$$C = \sum_{\langle i,j \rangle} \tfrac{1}{2}(1 - z_i z_j)$$



Goal: find a **bipartition** of vertices that cut the maximum # edges

Combinatorial Optimization Problems

Cost function

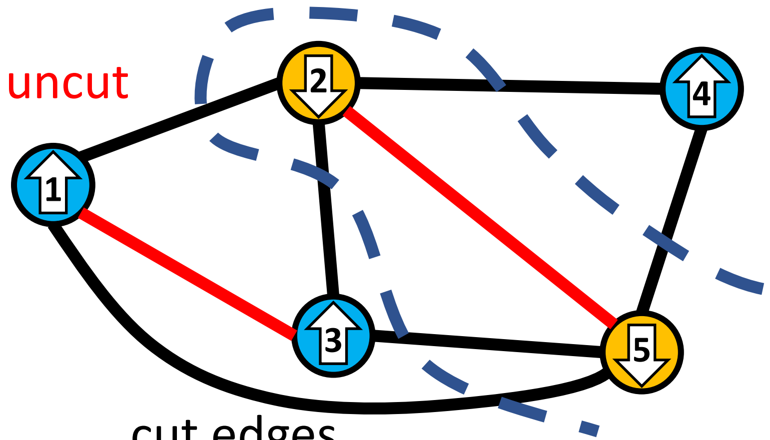
$$C(\boldsymbol{z}) = \sum_{\alpha} C_{\alpha}(\boldsymbol{z})$$

$$\boldsymbol{z} = (z_1, \ldots z_n) \in \{\pm 1\}^n$$

Want $\boldsymbol{z}^*$ **so** $C(\boldsymbol{z}^*)$ **is maximized**

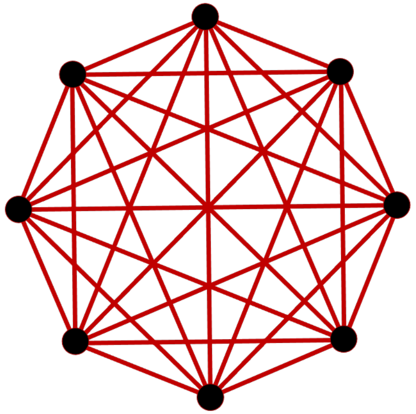
MaxCut

$$C = \sum_{\langle i,j \rangle} \frac{1}{2}(1 - z_i z_j)$$



Goal: find a **bipartition** of vertices that cut the maximum # edges

Sherrington-Kirkpatrick (SK) model



Combinatorial Optimization Problems

Cost function

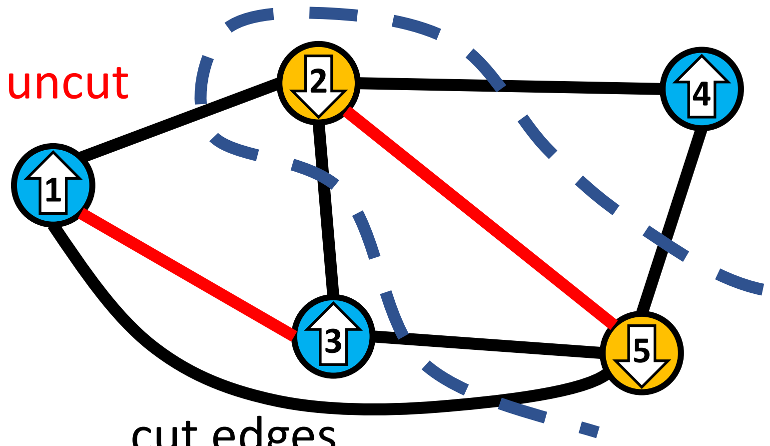
$$C(\boldsymbol{z}) = \sum_{\alpha} C_\alpha(\boldsymbol{z})$$

$$\boldsymbol{z} = (z_1, \dots z_n) \in \{\pm 1\}^n$$

Want $\boldsymbol{z}^*$ **so** $C(\boldsymbol{z}^*)$ **is maximized**

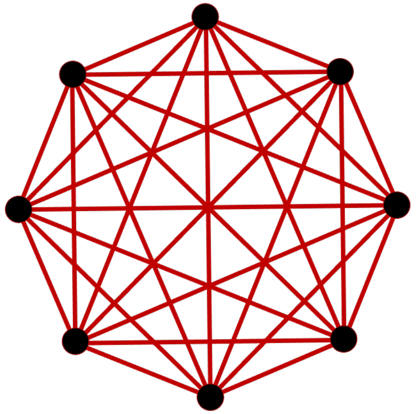
MaxCut

$$C = \sum_{\langle i,j \rangle} \tfrac{1}{2}(1 - z_i z_j)$$



Goal: find a **bipartition** of vertices that cut the maximum # edges

Sherrington-Kirkpatrick (SK) model



$$C = \frac{1}{\sqrt{n}} \sum_{i<j} J_{ij} z_i z_j$$

$$J_{ij} \sim \mathrm{Normal}(0,1)$$

or $J_{ij} \in \{\pm 1\}$

Combinatorial Optimization Problems

Cost function

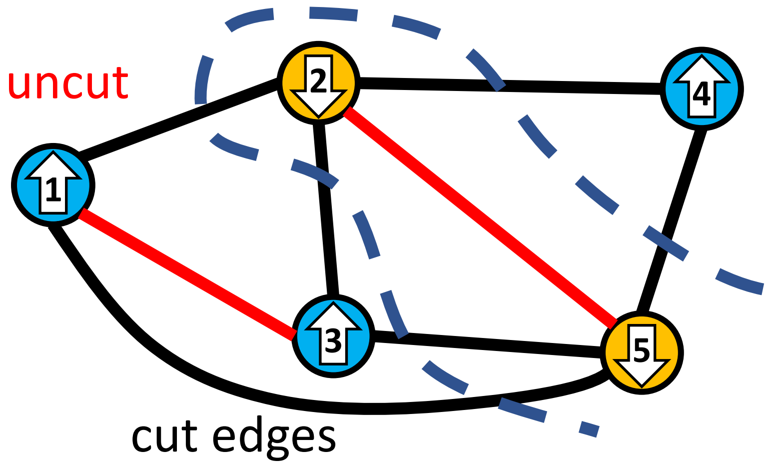
$$C(\boldsymbol{z}) = \sum_{\alpha} C_\alpha(\boldsymbol{z})$$

$$\boldsymbol{z} = (z_1, \ldots z_n) \in \{\pm 1\}^n$$

Want $\boldsymbol{z}^*$ **so** $C(\boldsymbol{z}^*)$ **is maximized**

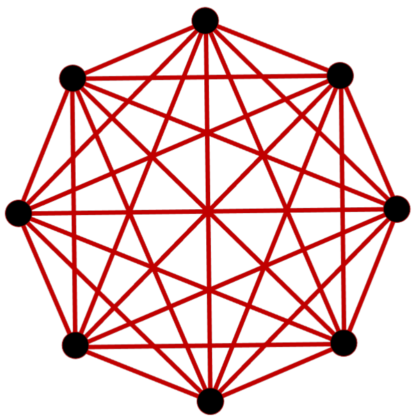
MaxCut

$$C = \sum_{\langle i,j \rangle} \tfrac{1}{2}(1 - z_i z_j)$$



Goal: find a **bipartition** of vertices that cut the maximum # edges

Sherrington-Kirkpatrick (SK) model



$$C = \frac{1}{\sqrt{n}} \sum_{i<j} J_{ij} z_i z_j$$

$$J_{ij} \sim \mathrm{Normal}(0,1)$$

or $J_{ij} \in \{\pm 1\}$

MaxCut on Erdős-Rényi graphs = SK (average case)

Quantum Approximate Optimization Algorithm (QAOA)

[Farhi Goldstone Gutmann 2014]

Quantum Approximate Optimization Algorithm (QAOA)

$|+\rangle$

$|+\rangle$

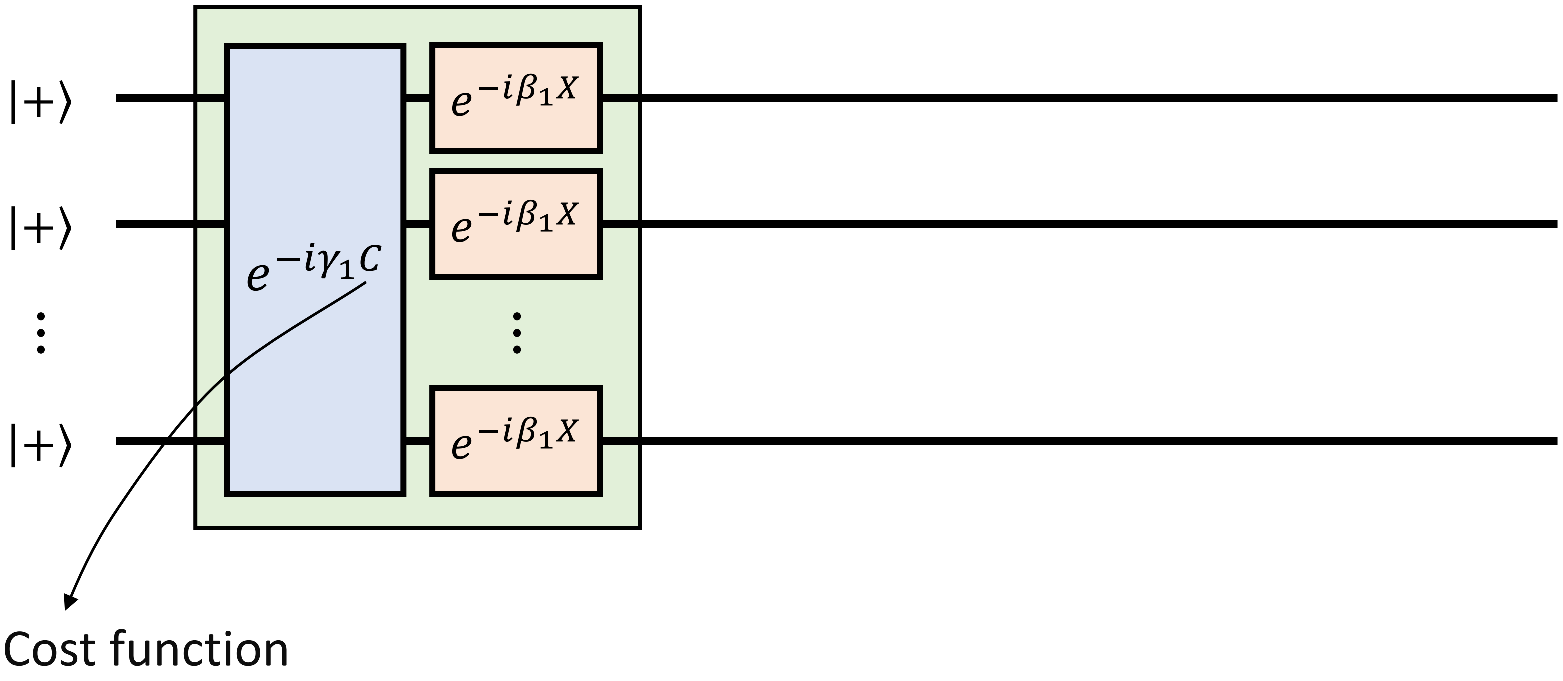
$\vdots$

$|+\rangle$

$$|+\rangle^{\otimes n}$$

[Farhi Goldstone Gutmann 2014]

Quantum Approximate Optimization Algorithm (QAOA)

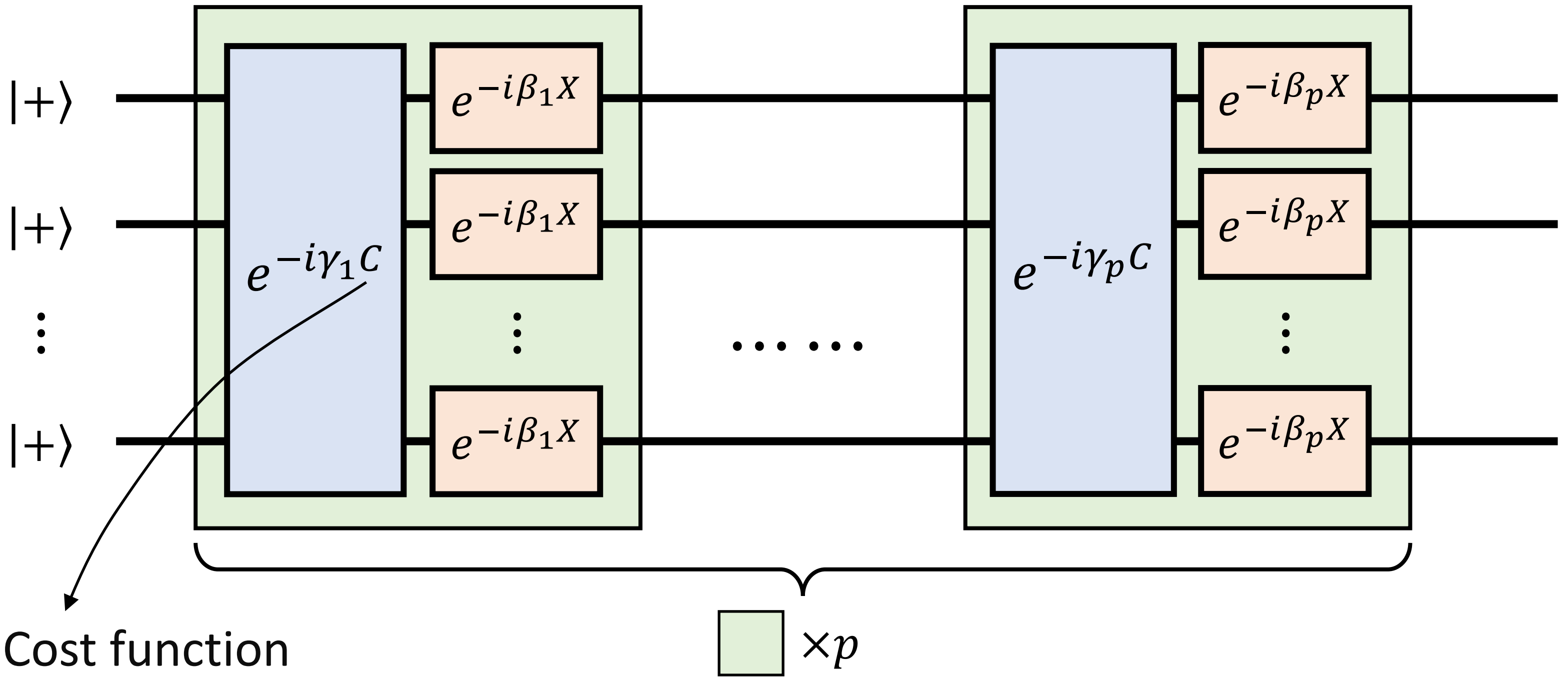


$$e^{-i\beta_1 B} e^{-i\gamma_1 C} \ |+\rangle^{\otimes n}$$

$$B = \sum_{i=1}^{n} X_i$$

[Farhi Goldstone Gutmann 2014]

Quantum Approximate Optimization Algorithm (QAOA)

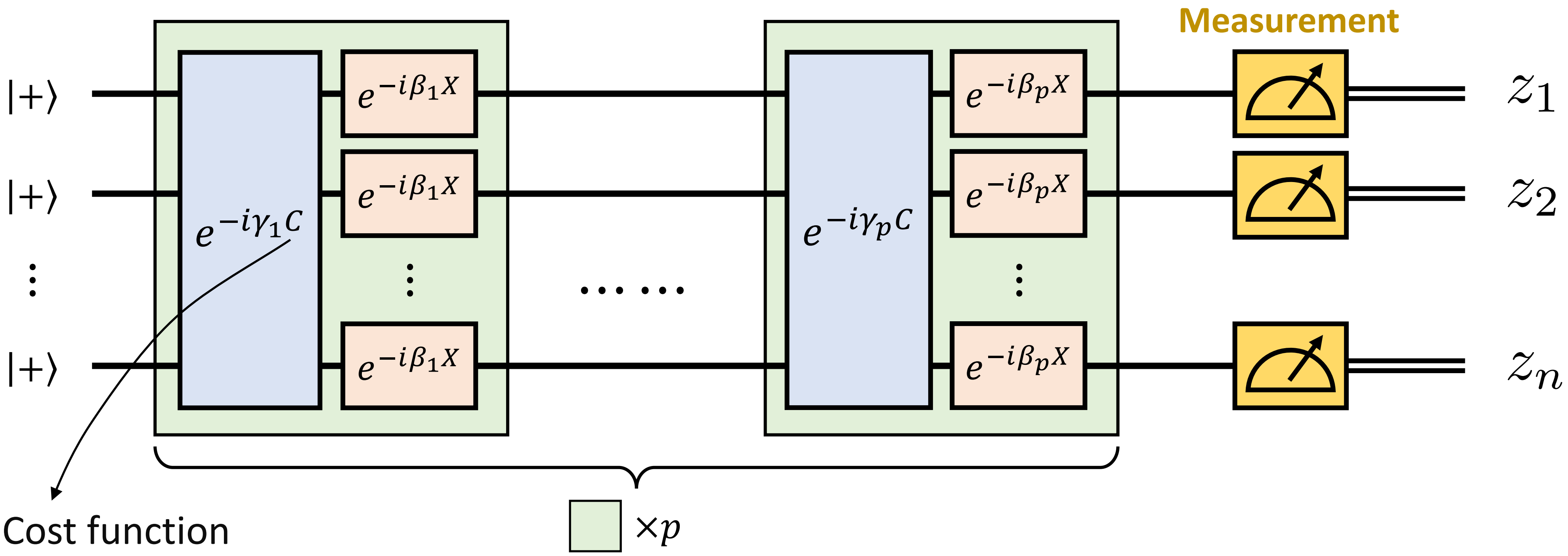


$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = e^{-i\beta_p B} e^{-i\gamma_p C} \cdots e^{-i\beta_1 B} e^{-i\gamma_1 C} \; |+\rangle^{\otimes n}$$

$$B = \sum_{i=1}^{n} X_i$$

[Farhi Goldstone Gutmann 2014]

Quantum Approximate Optimization Algorithm (QAOA)

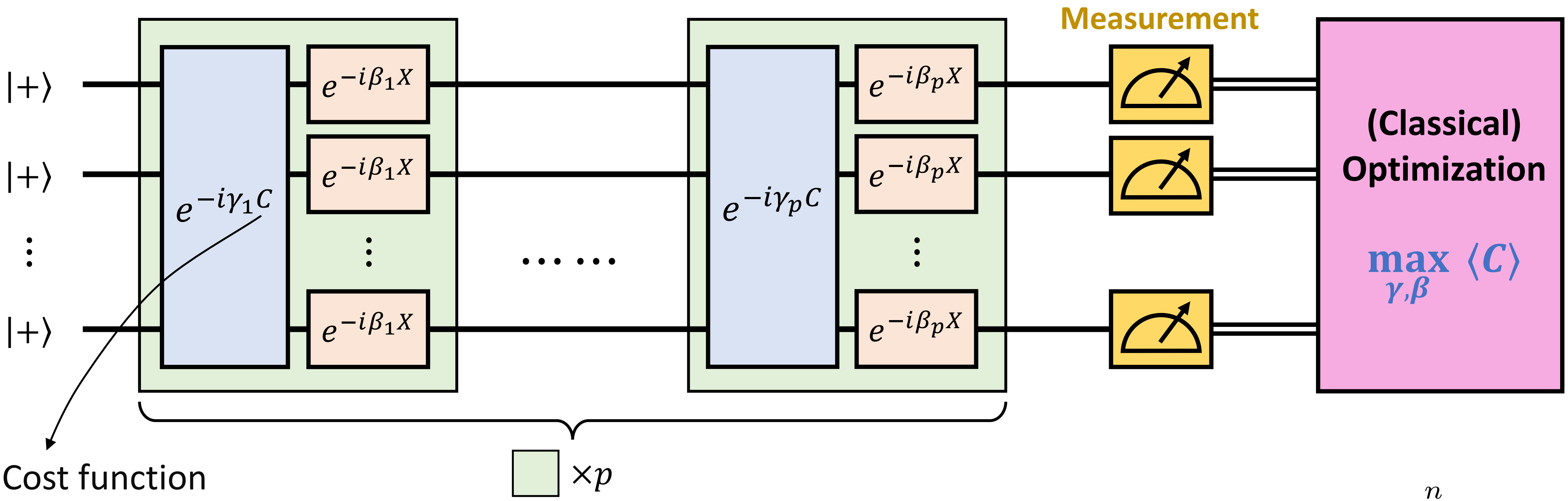


$$|\gamma, \boldsymbol{\beta}\rangle = e^{-i\beta_p B} e^{-i\gamma_p C} \ldots e^{-i\beta_1 B} e^{-i\gamma_1 C} \; |+\rangle^{\otimes n}$$

$$B = \sum_{i=1}^{n} X_i$$

[Farhi Goldstone Gutmann 2014]

Quantum Approximate Optimization Algorithm (QAOA)

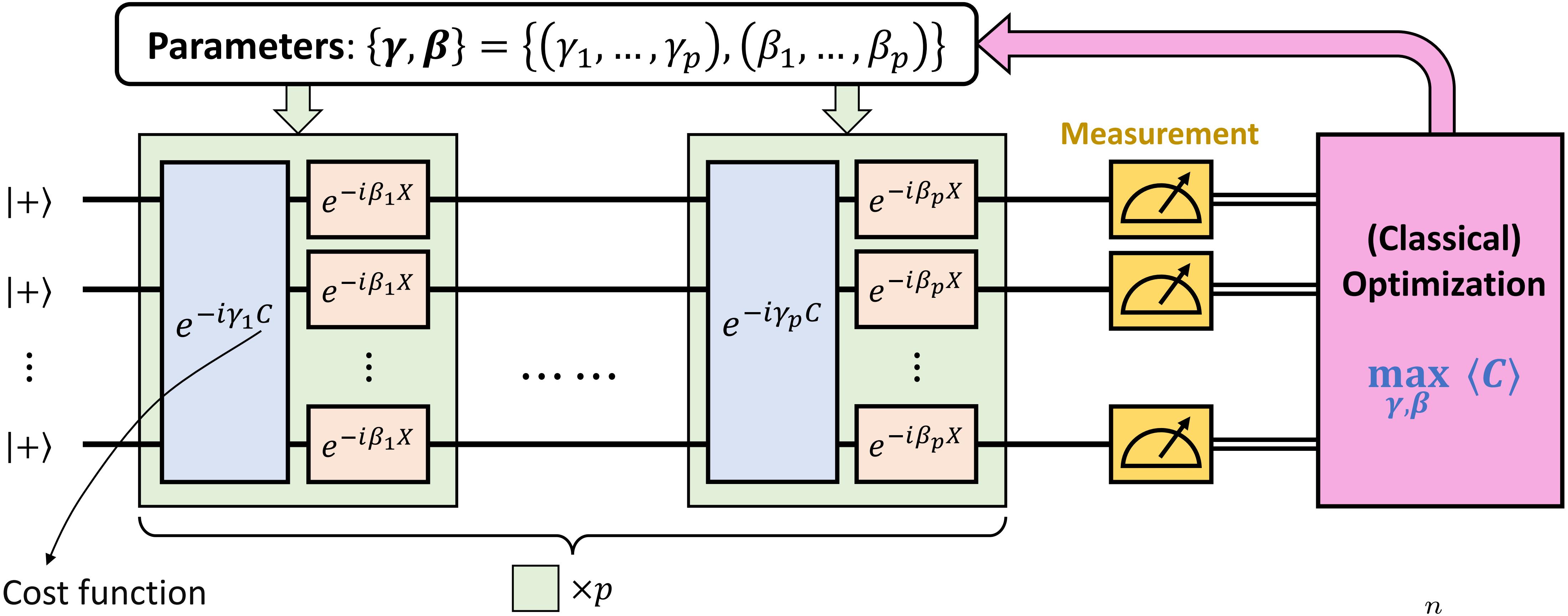


$$|\gamma, \boldsymbol{\beta}\rangle = e^{-i\beta_p B} e^{-i\gamma_p C} \dots e^{-i\beta_1 B} e^{-i\gamma_1 C} \; |+\rangle^{\otimes n}$$

$$B = \sum_{i=1}^{n} X_i$$

[Farhi Goldstone Gutmann 2014]

Quantum Approximate Optimization Algorithm (QAOA)

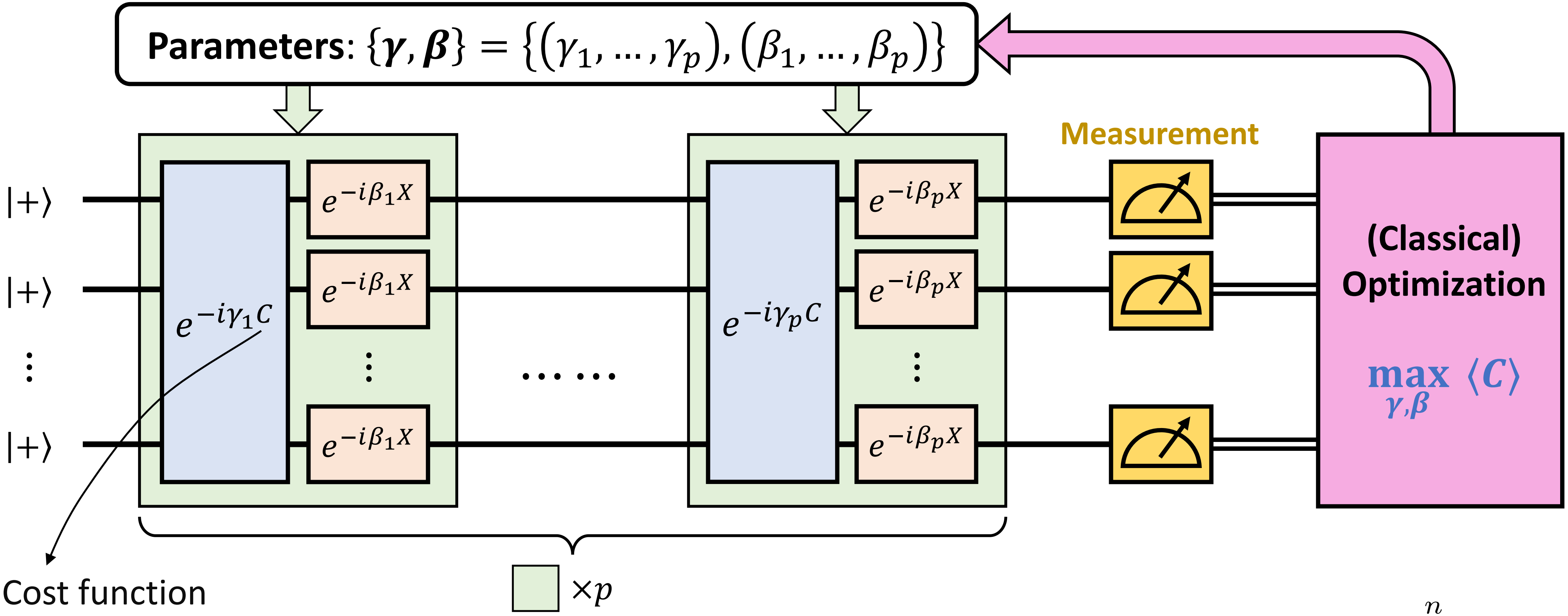


$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = e^{-i\beta_p B} e^{-i\gamma_p C} \ldots e^{-i\beta_1 B} e^{-i\gamma_1 C} \; |+\rangle^{\otimes n}$$

$$B = \sum_{i=1}^{n} X_i$$

[Farhi Goldstone Gutmann 2014]

Quantum Approximate Optimization Algorithm (QAOA)



$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = e^{-i\beta_p B} e^{-i\gamma_p C} \ldots e^{-i\beta_1 B} e^{-i\gamma_1 C} \; |+\rangle^{\otimes n}$$

$$B = \sum_{i=1}^{n} X_i$$

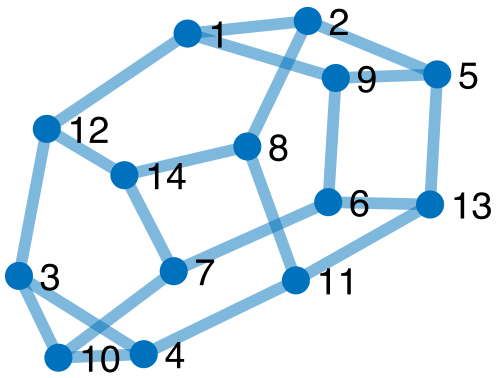
As $p \to \infty$ QAOA can get the global optimum

[Farhi Goldstone Gutmann 2014]

Previous Results on the QAOA

- Analyze performance via “subgraphs”

e.g. MaxCut on 3-regular graphs



$$C = \sum_{\langle j,k \rangle} \frac{1}{2}(1 - Z_j Z_k)$$

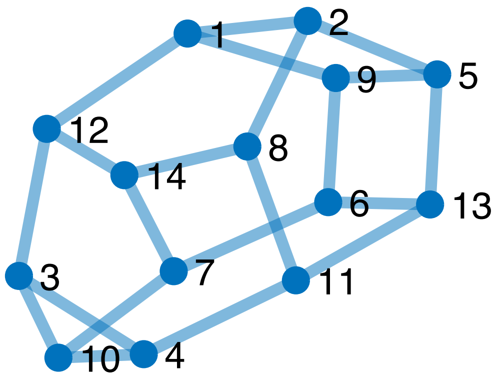
Previous Results on the QAOA

- Analyze performance via “subgraphs”

e.g. MaxCut on 3-regular graphs

$$p = 1$$

$$C = \sum_{\langle j,k \rangle} \frac{1}{2}(1 - Z_j Z_k)$$



Previous Results on the QAOA

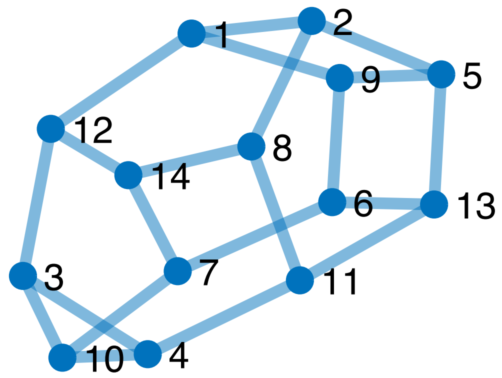
- Analyze performance via “subgraphs”

$$p = 1$$

$$|s\rangle = |+\rangle^{\otimes n}$$

e.g. MaxCut on 3-regular graphs

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

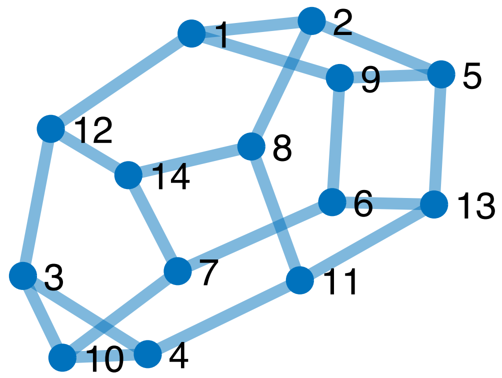


Previous Results on the QAOA

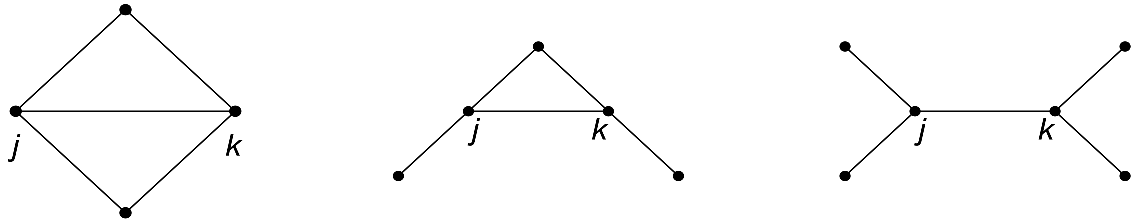
- Analyze performance via “subgraphs”

e.g. MaxCut on 3-regular graphs

$$p = 1 \qquad |s\rangle = |+\rangle^{\otimes n}$$



$$\langle s| \underbrace{e^{i\gamma C} e^{i\beta B} Z_j Z_k e^{-i\beta B} e^{-i\gamma C}}_{\text{supported on 3 types of subgraphs}} |s\rangle$$

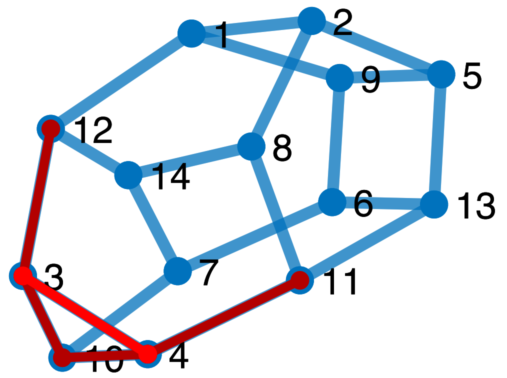


Previous Results on the QAOA

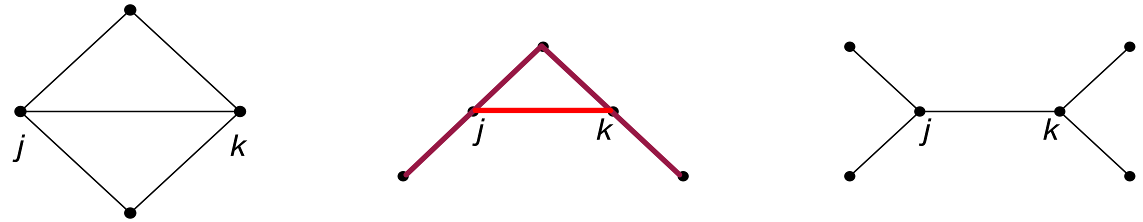
- Analyze performance via "subgraphs"

e.g. MaxCut on 3-regular graphs

$$p = 1 \qquad |s\rangle = |+\rangle^{\otimes n}$$



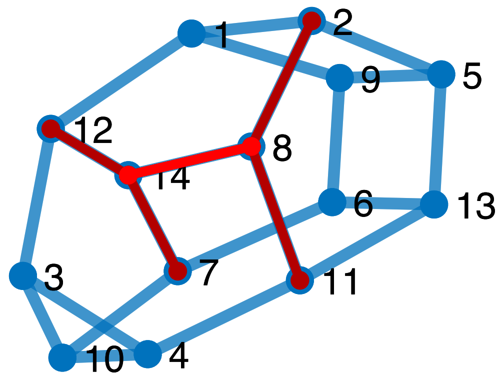
$$\langle s|\underbrace{e^{i\gamma C}e^{i\beta B}Z_jZ_k e^{-i\beta B}e^{-i\gamma C}}_{\text{supported on 3 types of subgraphs}}|s\rangle$$



Previous Results on the QAOA

- Analyze performance via “subgraphs”

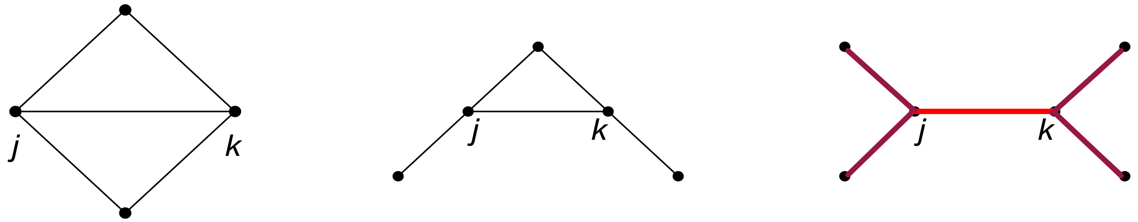
e.g. MaxCut on 3-regular graphs



$$p = 1 \qquad |s\rangle = |+\rangle^{\otimes n}$$

$$\langle s|\underbrace{e^{i\gamma C}e^{i\beta B}Z_jZ_ke^{-i\beta B}e^{-i\gamma C}}_{}|s\rangle$$

supported on 3 types of subgraphs

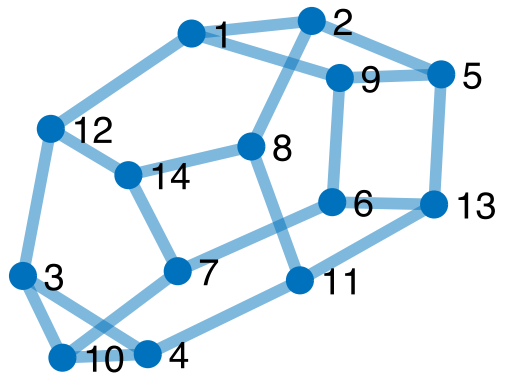


Previous Results on the QAOA

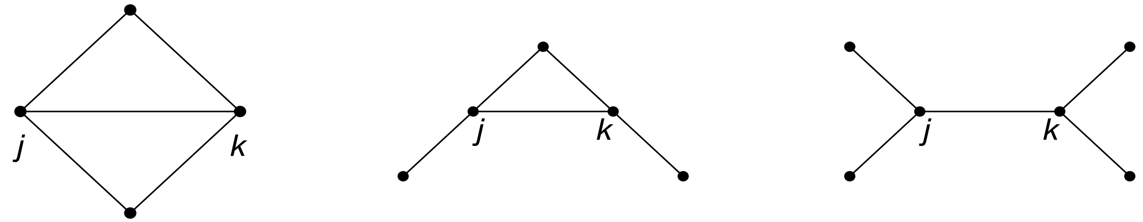
- Analyze performance via "subgraphs"

e.g. MaxCut on 3-regular graphs

$$p = 1 \qquad |s\rangle = |+\rangle^{\otimes n}$$



$$\langle s|\underbrace{e^{i\gamma C}e^{i\beta B}Z_jZ_ke^{-i\beta B}e^{-i\gamma C}}_{\text{supported on 3 types of subgraphs}}|s\rangle$$



Worst case guarantee:

$$\langle C\rangle / C_{\max} \geq 0.6924 \text{ @ } p = 1$$

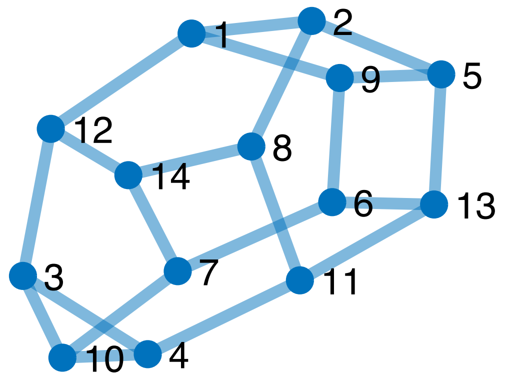
[Farhi Goldstone Gutmann 2014]

Previous Results on the QAOA

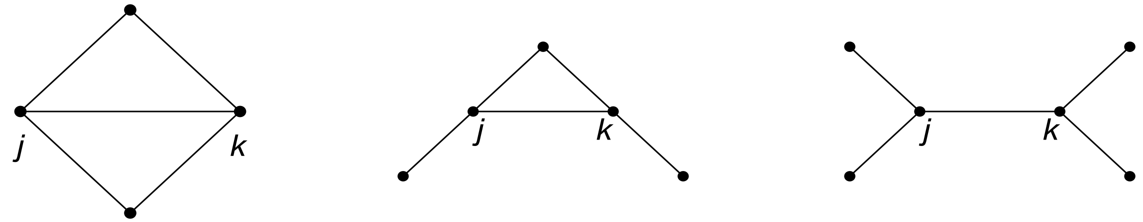
- Analyze performance via "subgraphs"

e.g. MaxCut on 3-regular graphs

$$p = 1 \qquad |s\rangle = |+\rangle^{\otimes n}$$



$$\langle s|\underbrace{e^{i\gamma C} e^{i\beta B} Z_j Z_k e^{-i\beta B} e^{-i\gamma C}}_{\text{supported on 3 types of subgraphs}}|s\rangle$$



Worst case guarantee:

$$\langle C\rangle / C_{\max} \geq 0.6924 \text{ @ } p = 1$$

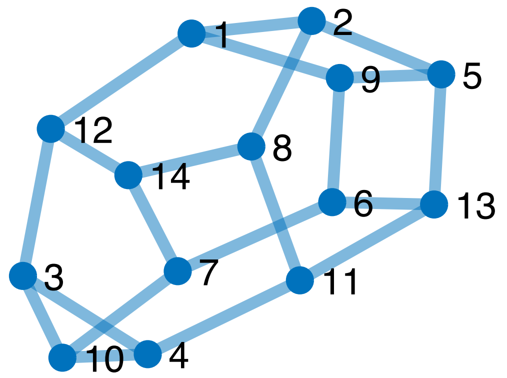
[Farhi Goldstone Gutmann 2014]

Difficult for higher p as the complexity of classical simulation grow as $O\left(2^{2^p}\right)$!

Previous Results on the QAOA

- Analyze performance via “subgraphs”

e.g. MaxCut on 3-regular graphs



“Landscape-Independence”

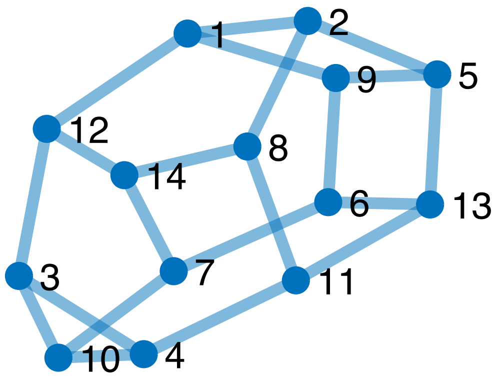
$$F_G(\boldsymbol{\gamma},\boldsymbol{\beta}) = \langle \boldsymbol{\gamma},\boldsymbol{\beta}|_G \, C_G \, |\boldsymbol{\gamma},\boldsymbol{\beta}\rangle_G$$

[**LZ** *et al.* 2018]
[*Brandão et al.* 2018]

Previous Results on the QAOA

- Analyze performance via "subgraphs"

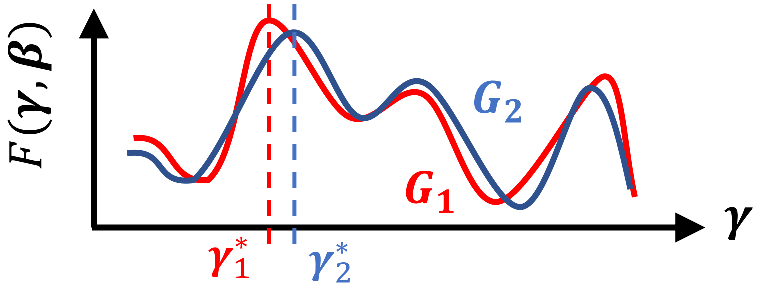
e.g. MaxCut on 3-regular graphs



"Landscape-Independence"

$$F_G(\boldsymbol{\gamma},\boldsymbol{\beta}) = \langle \boldsymbol{\gamma},\boldsymbol{\beta}|_G \, C_G \, |\boldsymbol{\gamma},\boldsymbol{\beta}\rangle_G$$

$$\xrightarrow{\quad n\to\infty \quad} F(\boldsymbol{\gamma},\boldsymbol{\beta}) \times \left[1 + O\left(\frac{1}{\sqrt{n}}\right)\right]$$

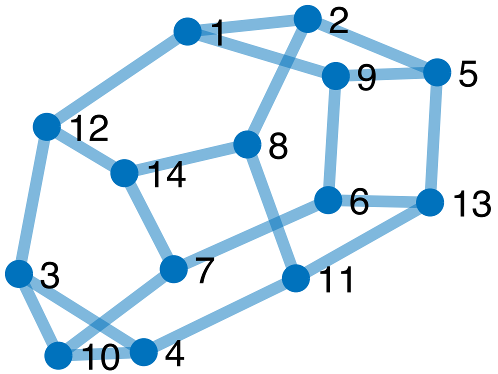


[**LZ** *et al.* 2018]
[*Brandão et al.* 2018]

Previous Results on the QAOA

- Analyze performance via “subgraphs”

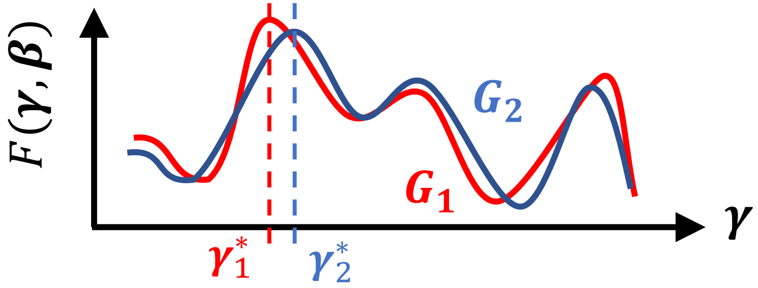
e.g. MaxCut on 3-regular graphs



“Landscape-Independence”

$$F_G(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \boldsymbol{\gamma}, \boldsymbol{\beta}|_G \, C_G \, |\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle_G$$

$$\xrightarrow{n \to \infty} F(\boldsymbol{\gamma}, \boldsymbol{\beta}) \times \left[1 + O\left(\frac{1}{\sqrt{n}}\right)\right]$$



[**LZ** *et al.* 2018]
[*Brandão et al.* 2018]

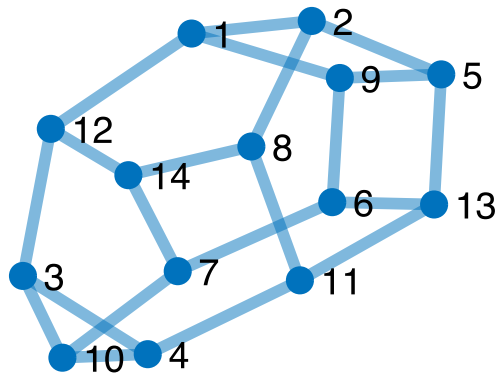
- Low-depth QAOA don’t see the whole graph → limited performance

[Bravyi Kliesch Koenig Tang 2019]
[Farhi Gamarnik Gutmann 2020]

Previous Results on the QAOA

- Analyze performance via "subgraphs"

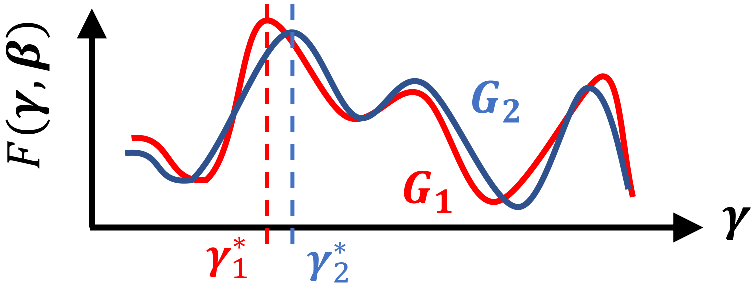
e.g. MaxCut on 3-regular graphs



"Landscape-Independence"

$$F_G(\boldsymbol{\gamma},\boldsymbol{\beta}) = \langle \boldsymbol{\gamma},\boldsymbol{\beta}|_G \, C_G \, |\boldsymbol{\gamma},\boldsymbol{\beta}\rangle_G$$

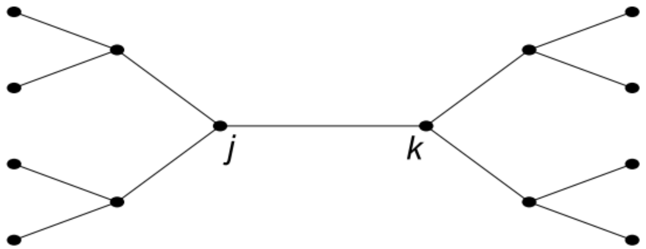
$$\xrightarrow{n\to\infty} F(\boldsymbol{\gamma},\boldsymbol{\beta}) \times \left[1 + O\left(\frac{1}{\sqrt{n}}\right)\right]$$



[**LZ** *et al.* 2018]
[*Brandão et al.* 2018]

- Low-depth QAOA don't see the whole graph → limited performance

On d-regular graphs, mostly see trees when $p \ll \log_{d-1} n$

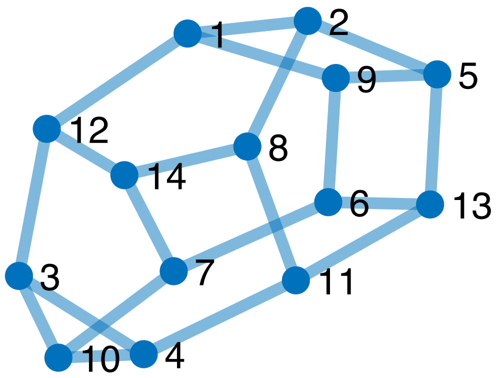


[Bravyi Kliesch Koenig Tang 2019]
[Farhi Gamarnik Gutmann 2020]

Previous Results on the QAOA

- Analyze performance via "subgraphs"

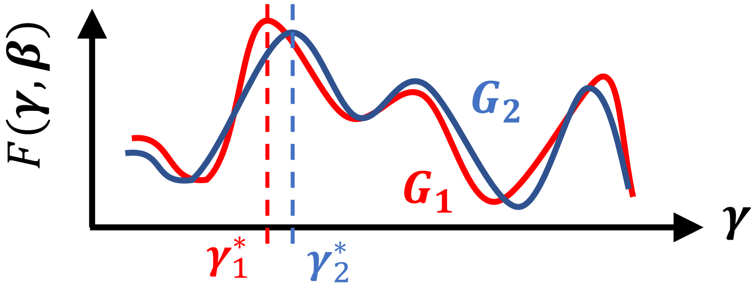
e.g. MaxCut on 3-regular graphs



"Landscape-Independence"

$$F_G(\boldsymbol{\gamma},\boldsymbol{\beta}) = \langle \boldsymbol{\gamma},\boldsymbol{\beta}|_G \, C_G \, |\boldsymbol{\gamma},\boldsymbol{\beta}\rangle_G$$

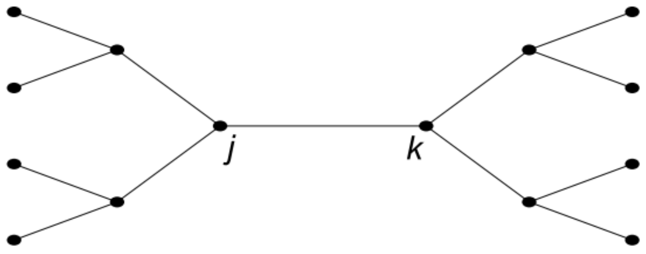
$$\xrightarrow{n\to\infty} F(\boldsymbol{\gamma},\boldsymbol{\beta}) \times \left[1 + O\left(\frac{1}{\sqrt{n}}\right)\right]$$



[**LZ** *et al.* 2018]
[*Brandão et al.* 2018]

- Low-depth QAOA don't see the whole graph → limited performance

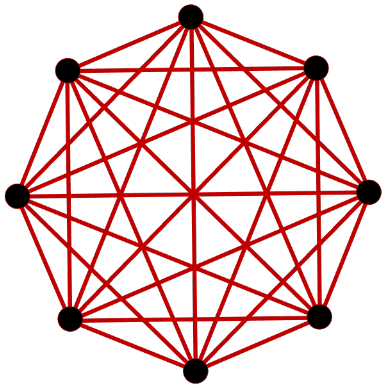
On d-regular graphs, mostly see trees when
$p \ll \log_{d-1} n$



[Bravyi Kliesch Koenig Tang 2019]
[Farhi Gamarnik Gutmann 2020]

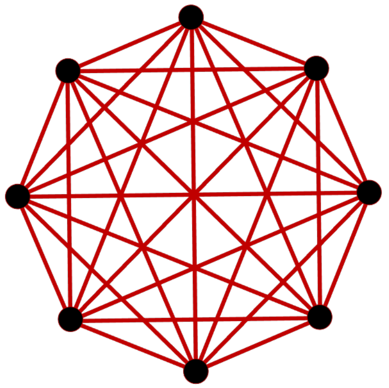
Cannot distinguish bipartite vs. typical (frustrated) graphs

The Sherrington-Kirkpatrick model



$$C_J = \frac{1}{\sqrt{n}} \sum_{i<j} J_{ij} Z_i Z_j \qquad J_{ij} \sim \mathrm{Normal}(0,1)$$

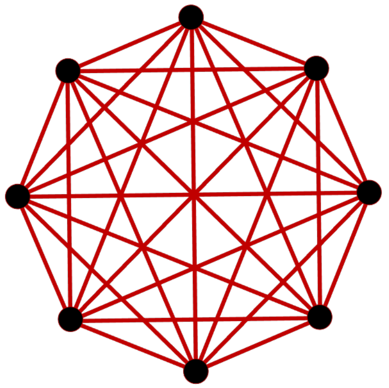
The Sherrington-Kirkpatrick model



$$C_J = \frac{1}{\sqrt{n}} \sum_{i<j} J_{ij} Z_i Z_j \qquad J_{ij} \sim \mathrm{Normal}(0,1)$$

- Unbounded vertex degree → QAOA sees the whole graph at $p = 2$

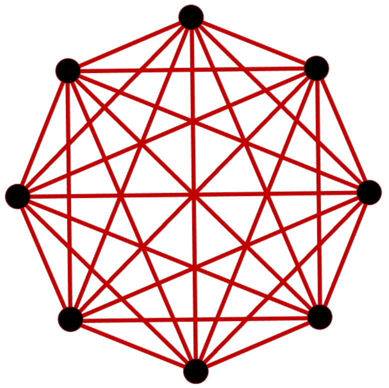
The Sherrington-Kirkpatrick model



$$C_J = \frac{1}{\sqrt{n}} \sum_{i<j} J_{ij} Z_i Z_j \qquad J_{ij} \sim \mathrm{Normal}(0,1)$$

- Unbounded vertex degree → QAOA sees the whole graph at $p = 2$
- **Worst case**: NP-hard to approximate within $O(1/\log^{c}(n))$ factor [Arora *et al.* 2005]

The Sherrington-Kirkpatrick model



$$C_J = \frac{1}{\sqrt{n}} \sum_{i<j} J_{ij} Z_i Z_j \qquad J_{ij} \sim \mathrm{Normal}(0,1)$$

- Unbounded vertex degree → QAOA sees the whole graph at $p = 2$

- **Worst case**: NP-hard to approximate within $O(1/\log^c(n))$ factor [Arora *et al.* 2005]
- **Typical case**: Famously, Parisi (1979) predicted and Talagrand (2006) proved that

$$\lim_{n\to\infty} \frac{1}{n} \max_{\boldsymbol{z}} C_J(\boldsymbol{z}) = \Pi_* = 0.763166\ldots$$

Complexity of solving a **typical** SK instance?

- Parisi *et al.*'s result does not *construct* the solution!

$$\Pi_* = 0.763166\ldots$$

Complexity of solving a **typical** SK instance?

- Parisi *et al.*'s result does not *<u>construct</u>* the solution!

$\Pi_* = 0.763166\ldots$

- Known results of typical-case complexity:

1. Simulated Annealing is believed to fail for this problem [Parisi]
2. Semi-Definite Programming obtains $C/n = 2/\pi \approx 0.6366$ [Montanari Sen 2016]
3. Assuming the conjecture that the SK model has no "***overlap gap property***" (OGP), Andrea Montanari's algorithm (2018) outputs $\hat{\boldsymbol{z}}$ with

$$C/n \geq (1-\epsilon)\Pi_* \quad \text{in time} \quad O(n^2/\epsilon^k)$$

Main Result 1:
Performance of the QAOA applied to the SK model

We give an $O(16^p)$-time method to evaluate

$$V_p(\boldsymbol{\gamma},\boldsymbol{\beta}) = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}_J[\langle \boldsymbol{\gamma},\boldsymbol{\beta}|C|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle]$$

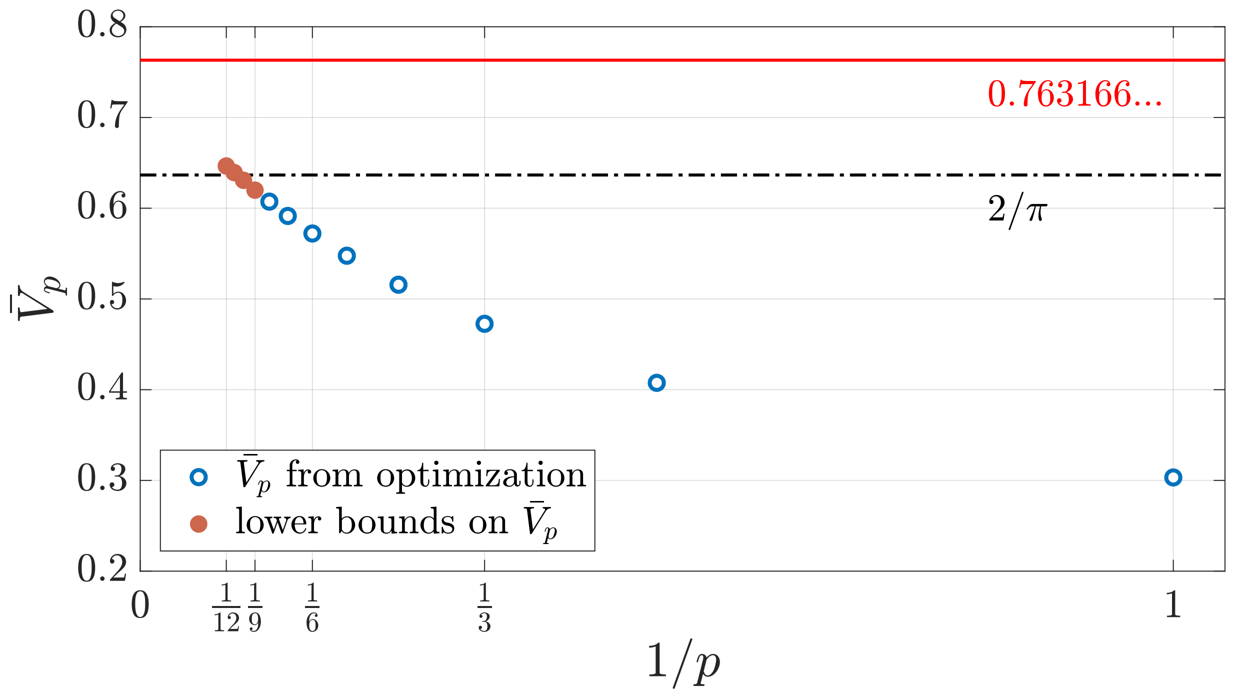
Much better than $O(2^{2^p})$-time subgraph method

Main Result 1:
Performance of the QAOA applied to the SK model

We give an $O(16^p)$-time method to evaluate

$$V_p(\boldsymbol{\gamma},\boldsymbol{\beta}) = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}_J[\langle\boldsymbol{\gamma},\boldsymbol{\beta}|C|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle]$$

Much better than $O(2^{2^p})$-time subgraph method



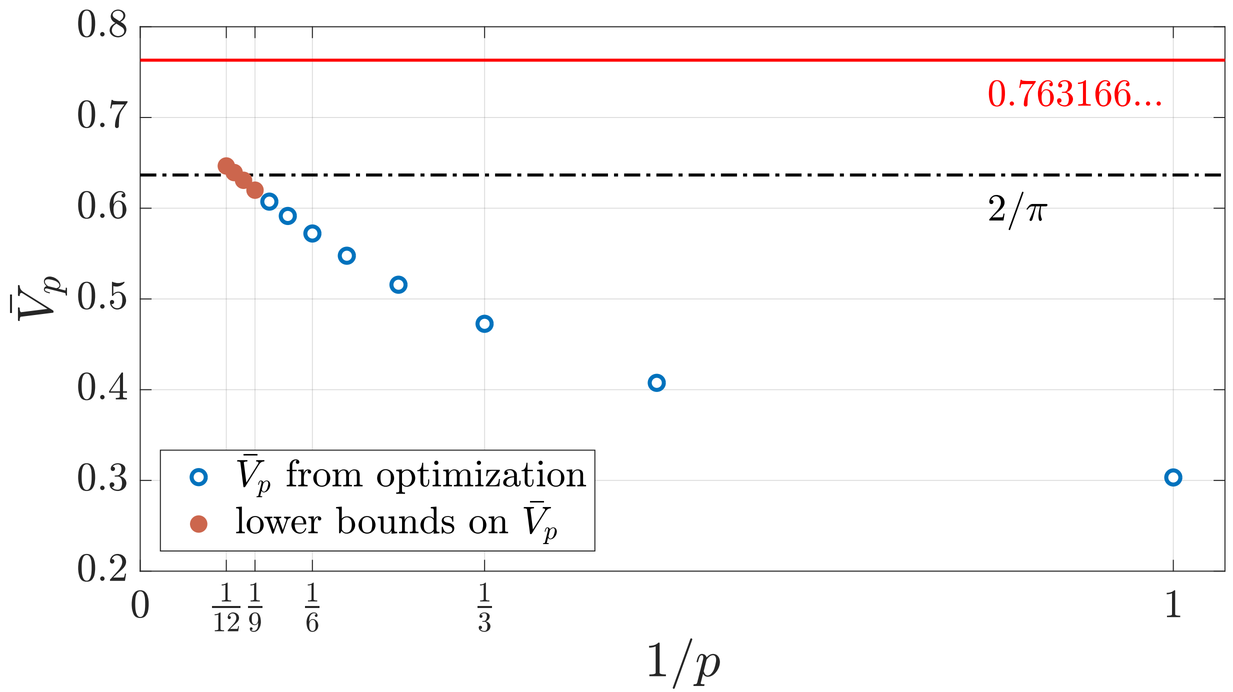
$$\bar{V}_p = \max_{\boldsymbol{\gamma},\boldsymbol{\beta}} V_p(\boldsymbol{\gamma},\boldsymbol{\beta})$$

Main Result 1:
Performance of the QAOA applied to the SK model

We give an $O(16^p)$-time method to evaluate

$$V_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \lim_{n \to \infty} \frac{1}{n} \mathbb{E}_J[\langle \boldsymbol{\gamma}, \boldsymbol{\beta} | C | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle]$$

Much better than $O(2^{2^p})$-time subgraph method



$$\bar{V}_p = \max_{\boldsymbol{\gamma}, \boldsymbol{\beta}} V_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$$

QAOA beats SDP @ *p*=11

Main Result 2:

Concentration of QAOA on the SK model

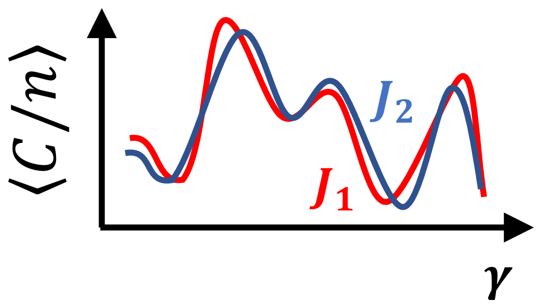
- We also prove, for any fixed depth p:

$$\lim_{n\to\infty} \mathbb{E}_J[\langle (C/n)^2 \rangle] = \lim_{n\to\infty} \mathbb{E}_J^2[\langle C/n \rangle]$$

Main Result 2:
Concentration of QAOA on the SK model

- We also prove, for any fixed depth p: $\lim_{n\to\infty} \mathbb{E}_J[\langle (C/n)^2 \rangle] = \lim_{n\to\infty} \mathbb{E}_J^2[\langle C/n \rangle]$

Concentration over instances
("Landscape-Independence")

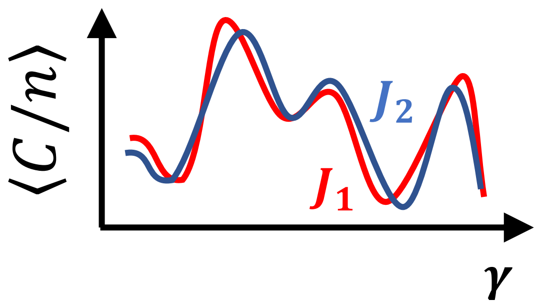


Main Result 2:
Concentration of QAOA on the SK model

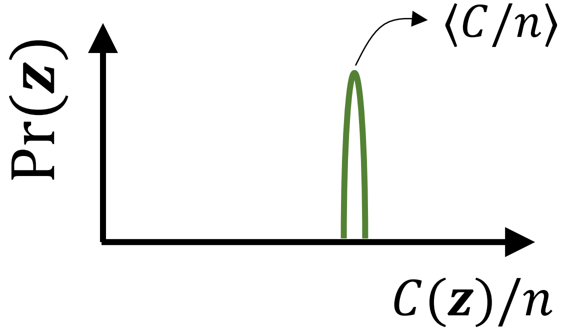
- We also prove, for any fixed depth p:

$$\lim_{n\to\infty} \mathbb{E}_J[\langle (C/n)^2 \rangle] = \lim_{n\to\infty} \mathbb{E}_J^2[\langle C/n \rangle]$$

Concentration over instances
("Landscape-Independence")



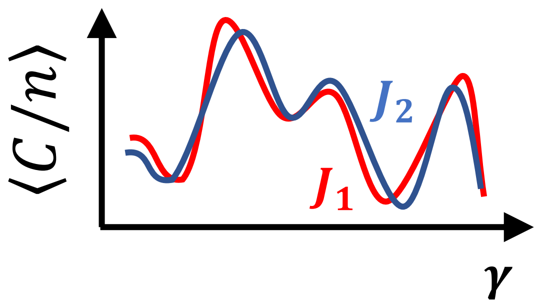
Concentration over measurements



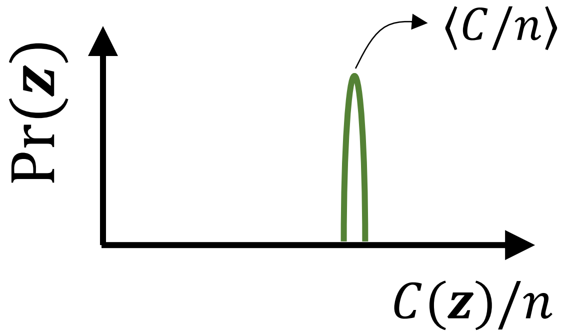
Main Result 2:
Concentration of QAOA on the SK model

- We also prove, for any fixed depth p: $\lim_{n\to\infty} \mathbb{E}_J[\langle (C/n)^2 \rangle] = \lim_{n\to\infty} \mathbb{E}_J^2[\langle C/n \rangle]$

Concentration over instances
("Landscape-Independence")



Concentration over measurements



- With probability $\to 1$ as $n \to \infty$, applying QAOA and measuring will give us a bit string $\boldsymbol{z}$ which has

$$C(\boldsymbol{z})/n \approx \langle C/n \rangle \approx V_p$$

Key Idea: Average over instances

- Parisi's formalism requires delicate tricks
  - A replica-symmetry-breaking ansatz for the free energy:

$$\mathbb{E}_J[\log Z_J]$$

$$Z_J(T) = \mathrm{tr}(e^{C_J/T})$$

Key Idea: **Average over instances**

- Parisi's formalism requires delicate tricks
  - A replica-symmetry-breaking ansatz for the free energy:

$$\mathbb{E}_J[\log Z_J] \neq \log \mathbb{E}_J[Z_J]$$

$$Z_J(T) = \mathrm{tr}(e^{C_J/T})$$

Key Idea: Average over instances

- Parisi's formalism requires delicate tricks
  - A replica-symmetry-breaking ansatz for the free energy:

$$\mathbb{E}_J[\log Z_J] = \lim_{k \to 0} \frac{1}{k} \log \mathbb{E}[Z_J^k] \qquad Z_J(T) = \mathrm{tr}(e^{C_J/T})$$

k replicas of J

Key Idea: Average over instances

- Parisi's formalism requires delicate tricks
  - A replica-symmetry-breaking ansatz for the free energy:

$$\mathbb{E}_J[\log Z_J] = \lim_{k\to 0} \frac{1}{k} \log \mathbb{E}[Z_J^k] \qquad Z_J(T) = \mathrm{tr}(e^{C_J/T})$$

k replicas of J

- For QAOA, averaging over J is easier

$$\frac{1}{n}\mathbb{E}_J[\langle C\rangle] = \mathbb{E}_J\Big[\,\langle s|e^{i\gamma C}e^{i\beta B}\frac{C}{n}e^{-i\beta B}e^{-i\gamma C}|s\rangle\,\Big]$$

Key Idea: **Average over instances**

- Parisi's formalism requires delicate tricks
  - A replica-symmetry-breaking ansatz for the free energy:

$$\mathbb{E}_J[\log Z_J] = \lim_{k\to 0} \frac{1}{k} \log \mathbb{E}[Z_J^k] \qquad Z_J(T) = \mathrm{tr}(e^{C_J/T})$$

k replicas of J

- For QAOA, averaging over J is easier

$$\frac{1}{n}\mathbb{E}_J[\langle C\rangle] = \mathbb{E}_J\left[\langle s|e^{i\gamma C} e^{i\beta B}\frac{C}{n} e^{-i\beta B} e^{-i\gamma C}|s\rangle\right]$$

$$= \frac{1}{2^n}\sum_{\boldsymbol{z}^1 \boldsymbol{z}^m \boldsymbol{z}^2} \mathbb{E}_J\left[e^{i\gamma C(\boldsymbol{z}^1)}\langle \boldsymbol{z}^1|e^{i\beta B}|\boldsymbol{z}^m\rangle \frac{C(\boldsymbol{z}^m)}{n}\langle \boldsymbol{z}^m|e^{-i\beta B}|\boldsymbol{z}^2\rangle e^{-i\gamma C(\boldsymbol{z}^2)}\right]$$

Key Idea: **Average over instances**

- Parisi's formalism requires delicate tricks
  - A replica-symmetry-breaking ansatz for the free energy:

$$\mathbb{E}_J[\log Z_J] = \lim_{k\to 0}\frac{1}{k}\log\mathbb{E}[Z_J^k] \qquad Z_J(T) = \mathrm{tr}(e^{C_J/T})$$

k replicas of J

- For QAOA, averaging over J is easier

$$C = \frac{1}{\sqrt{n}}\sum_{i<j} J_{ij} Z_i Z_j$$

$$\frac{1}{n}\mathbb{E}_J[\langle C\rangle] = \mathbb{E}_J\left[\langle s|e^{i\gamma C}e^{i\beta B}\frac{C}{n}e^{-i\beta B}e^{-i\gamma C}|s\rangle\right]$$

$$= \frac{1}{2^n}\sum_{\boldsymbol{z}^1\boldsymbol{z}^m\boldsymbol{z}^2}\mathbb{E}_J\left[e^{i\gamma C(\boldsymbol{z}^1)}\langle\boldsymbol{z}^1|e^{i\beta B}|\boldsymbol{z}^m\rangle\frac{C(\boldsymbol{z}^m)}{n}\langle\boldsymbol{z}^m|e^{-i\beta B}|\boldsymbol{z}^2\rangle e^{-i\gamma C(\boldsymbol{z}^2)}\right]$$

Key Idea: **Average over instances**

- Parisi's formalism requires delicate tricks
  - A replica-symmetry-breaking ansatz for the free energy:

$$\mathbb{E}_J[\log Z_J] = \lim_{k\to 0} \frac{1}{k} \log \mathbb{E}[Z_J^k] \qquad Z_J(T) = \mathrm{tr}(e^{C_J/T})$$

k replicas of J

- For QAOA, averaging over J is easier

$$C = \frac{1}{\sqrt{n}} \sum_{i<j} J_{ij} Z_i Z_j$$

$$\frac{1}{n}\mathbb{E}_J[\langle C\rangle] = \mathbb{E}_J\Big[\,\langle s|e^{i\gamma C} e^{i\beta B} \frac{C}{n} e^{-i\beta B} e^{-i\gamma C}|s\rangle\,\Big]$$

$$= \frac{1}{2^n} \sum_{\boldsymbol{z}^1 \boldsymbol{z}^m \boldsymbol{z}^2} \mathbb{E}_J\left[e^{i\gamma C(\boldsymbol{z}^1)} \langle \boldsymbol{z}^1|e^{i\beta B}|\boldsymbol{z}^m\rangle \frac{C(\boldsymbol{z}^m)}{n} \langle \boldsymbol{z}^m|e^{-i\beta B}|\boldsymbol{z}^2\rangle\, e^{-i\gamma C(\boldsymbol{z}^2)}\right]$$

Key Idea: **Average over instances**

- Parisi's formalism requires delicate tricks
  - A replica-symmetry-breaking ansatz for the free energy:

$$\mathbb{E}_J[\log Z_J] = \lim_{k\to 0} \frac{1}{k} \log \mathbb{E}[Z_J^k]$$

$$Z_J(T) = \mathrm{tr}(e^{C_J/T})$$

k replicas of J

- For QAOA, averaging over J is easier

$$C = \frac{1}{\sqrt{n}} \sum_{i<j} J_{ij} Z_i Z_j$$

$$\frac{1}{n}\mathbb{E}_J[\langle C\rangle] = \mathbb{E}_J\Big[\langle s|e^{i\gamma C} e^{i\beta B} \frac{C}{n} e^{-i\beta B} e^{-i\gamma C}|s\rangle\Big]$$

$$= \frac{1}{2^n} \sum_{\boldsymbol{z}^1 \boldsymbol{z}^m \boldsymbol{z}^2} \mathbb{E}_J\Big[e^{i\gamma C(\boldsymbol{z}^1)} \langle \boldsymbol{z}^1|e^{i\beta B}|\boldsymbol{z}^m\rangle \frac{C(\boldsymbol{z}^m)}{n} \langle \boldsymbol{z}^m|e^{-i\beta B}|\boldsymbol{z}^2\rangle e^{-i\gamma C(\boldsymbol{z}^2)}\Big]$$

For ϕ small, use $\mathbb{E}_J[e^{iJ\phi}] = 1 - \frac{1}{2}\phi^2 + \cdots \qquad \mathbb{E}_J[Je^{iJ\phi}] = i\phi + \cdots$

Key Idea: Average over instances

$$\frac{1}{n}\mathbb{E}_J[\langle C\rangle] \approx \frac{i}{n^{3/2}} \sum_{\boldsymbol{z}^1,\boldsymbol{z}^2} \Big[\langle \boldsymbol{z}^1 | e^{i\beta B} | \mathbf{1}\rangle \langle \mathbf{1} | e^{-i\beta B} | \boldsymbol{z}^2\rangle \sum_{k<\ell} \phi_{k\ell} \prod_{i<j} \big(1 - \frac{1}{2}\phi_{ij}^2\big)\Big]$$

$$\phi_{ab} = \frac{\gamma}{\sqrt{n}}(z_a^1 z_b^1 - z_a^2 z_b^2)$$

Key Idea: Average over instances

$$\frac{1}{n}\mathbb{E}_J[\langle C\rangle] \approx \frac{i}{n^{3/2}} \sum_{\boldsymbol{z}^1,\boldsymbol{z}^2} \Big[\langle \boldsymbol{z}^1 | e^{i\beta B} | \mathbf{1}\rangle \langle \mathbf{1} | e^{-i\beta B} | \boldsymbol{z}^2 \rangle \sum_{k<\ell} \phi_{k\ell} \prod_{i<j} \big(1 - \frac{1}{2}\phi_{ij}^2\big)\Big]$$

Permutation symmetry → configuration basis

$$\phi_{ab} = \frac{\gamma}{\sqrt{n}}(z_a^1 z_b^1 - z_a^2 z_b^2)$$

$$\boldsymbol{z}^1 \quad + + + + + + + + - - - - - - -$$

$$\boldsymbol{z}^2 \quad + + + + - - - - + + + + - - -$$

Key Idea: **Average over instances**

$$\frac{1}{n}\mathbb{E}_J[\langle C\rangle] \approx \frac{i}{n^{3/2}} \sum_{\boldsymbol{z}^1,\boldsymbol{z}^2} \Big[\langle \boldsymbol{z}^1|e^{i\beta B}|\mathbf{1}\rangle \langle \mathbf{1}|e^{-i\beta B}|\boldsymbol{z}^2\rangle \sum_{k<\ell}\phi_{k\ell}\prod_{i<j}\big(1-\frac{1}{2}\phi_{ij}^2\big)\Big]$$

Permutation symmetry → configuration basis

$$\phi_{ab} = \frac{\gamma}{\sqrt{n}}(z_a^1 z_b^1 - z_a^2 z_b^2)$$

$$\begin{array}{lcccc}
\boldsymbol{z}^1 & + + + + & + + + + & - - - - & - - - \\
\boldsymbol{z}^2 & \underbrace{+ + + +}_{n_{++}} & \underbrace{- - - -}_{n_{+-}} & \underbrace{+ + + +}_{n_{-+}} & \underbrace{- - -}_{n_{--}}
\end{array}$$

$$\sum_{\boldsymbol{a}} n_{\boldsymbol{a}} = n$$

Key Idea: **Average over instances**

$$\frac{1}{n}\mathbb{E}_J[\langle C\rangle] \approx \frac{i}{n^{3/2}} \sum_{\boldsymbol{z}^1,\boldsymbol{z}^2} \left[\langle \boldsymbol{z}^1 | e^{i\beta B} | \mathbf{1} \rangle \langle \mathbf{1} | e^{-i\beta B} | \boldsymbol{z}^2 \rangle \sum_{k<\ell} \phi_{k\ell} \prod_{i<j} \left(1 - \frac{1}{2}\phi_{ij}^2\right) \right]$$

Permutation symmetry → configuration basis

$$\phi_{ab} = \frac{\gamma}{\sqrt{n}}(z_a^1 z_b^1 - z_a^2 z_b^2)$$

$$\begin{array}{c|c|c|c|c}
\boldsymbol{z}^1 & ++++ & ++++ & ---- & --- \\
\boldsymbol{z}^2 & ++++ & ---- & ++++ & --- \\
 & \underbrace{\qquad\qquad}_{n_{++}} & \underbrace{\qquad\qquad}_{n_{+-}} & \underbrace{\qquad\qquad}_{n_{-+}} & \underbrace{\qquad\qquad}_{n_{--}}
\end{array}$$

$$\sum_{\boldsymbol{a}} n_{\boldsymbol{a}} = n$$

$$\sum_{\boldsymbol{z}^1,\boldsymbol{z}^2} = \sum_{\{n_{\boldsymbol{a}}\}} \binom{n}{n_{++}, n_{+-}, n_{-+}, n_{--}}$$

n-bit strings

configurations

Key Idea: **Average over instances**

$$\frac{1}{n}\mathbb{E}_J[\langle C\rangle] \approx \frac{i}{n^{3/2}} \sum_{\boldsymbol{z}^1,\boldsymbol{z}^2} \left[\langle \boldsymbol{z}^1 | e^{i\beta B} | \mathbf{1}\rangle \langle \mathbf{1} | e^{-i\beta B} | \boldsymbol{z}^2 \rangle \sum_{k<\ell} \phi_{k\ell} \prod_{i<j} \left(1 - \frac{1}{2}\phi_{ij}^2\right) \right]$$

Permutation symmetry → configuration basis

$$\phi_{ab} = \frac{\gamma}{\sqrt{n}}(z_a^1 z_b^1 - z_a^2 z_b^2)$$

$$\begin{array}{lcccc}
\boldsymbol{z}^1 & + + + + & + + + + & - - - - & - - - \\
\boldsymbol{z}^2 & \underbrace{+ + + +}_{n_{++}} & \underbrace{- - - -}_{n_{+-}} & \underbrace{+ + + +}_{n_{-+}} & \underbrace{- - -}_{n_{--}}
\end{array}$$

$$\sum_{\boldsymbol{a}} n_{\boldsymbol{a}} = n$$

$$\sum_{\boldsymbol{z}^1,\boldsymbol{z}^2} = \sum_{\{n_{\boldsymbol{a}}\}} \binom{n}{n_{++}, n_{+-}, n_{-+}, n_{--}}$$

n-bit strings

configurations

For general p, there are 2^{2p} configurations

$\exp(O(p))$ complexity

Performance of the QAOA on the SK model

- Turn the crank, we get at $p = 1$

$$V_1 = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}_J[\langle C\rangle] = \gamma e^{-2\gamma^2} \sin 4\beta$$

Optimum @ $\beta = \frac{\pi}{8}, \gamma = \frac{1}{2}$

$$\Rightarrow \max_{\gamma,\beta} V_1 = \frac{1}{\sqrt{4e}} \approx 0.303$$

Performance of the QAOA on the SK model

- Turn the crank, we get at $p = 1$

$$V_1 = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}_J[\langle C\rangle] = \gamma e^{-2\gamma^2} \sin 4\beta$$

Optimum @ $\beta = \frac{\pi}{8}, \gamma = \frac{1}{2}$

$$\Rightarrow \max_{\gamma,\beta} V_1 = \frac{1}{\sqrt{4e}} \approx 0.303$$

- Can also show **concentration**

$$\lim_{n\to\infty} \mathbb{E}_J[\langle (C/n)^2\rangle] = \lim_{n\to\infty} \mathbb{E}_J^2[\langle C/n\rangle]$$

Performance of the QAOA on the SK model

- Turn the crank, we get at $p = 1$

$$V_1 = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}_J[\langle C\rangle] = \gamma e^{-2\gamma^2} \sin 4\beta$$

Optimum @ $\beta = \frac{\pi}{8}, \gamma = \frac{1}{2}$

$$\Rightarrow \max_{\gamma,\beta} V_1 = \frac{1}{\sqrt{4e}} \approx 0.303$$

- Can also show **concentration**

$$\lim_{n\to\infty} \mathbb{E}_J[\langle (C/n)^2\rangle] = \lim_{n\to\infty} \mathbb{E}_J^2[\langle C/n\rangle]$$

- Generic vs. optimized QAOA

Performance of the QAOA on the SK model

- Turn the crank, we get at $p = 1$

$$V_1 = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}_J[\langle C\rangle] = \gamma e^{-2\gamma^2} \sin 4\beta$$

Optimum @ $\beta = \frac{\pi}{8}, \gamma = \frac{1}{2}$

$$\Rightarrow \max_{\gamma,\beta} V_1 = \frac{1}{\sqrt{4e}} \approx 0.303$$

- Can also show **concentration**

$$\lim_{n\to\infty} \mathbb{E}_J[\langle (C/n)^2\rangle] = \lim_{n\to\infty} \mathbb{E}_J^2[\langle C/n\rangle]$$

- Generic vs. optimized QAOA

Suppose $J_{ij} \in \{\pm 1\}$

Then γ is periodic on $[0, \sqrt{n}\pi)$

β is periodic on $[0, \pi/2)$

Performance of the QAOA on the SK model

- Turn the crank, we get at $p = 1$

$$V_1 = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}_J[\langle C\rangle] = \gamma e^{-2\gamma^2} \sin 4\beta$$

Optimum @ $\beta = \frac{\pi}{8}, \gamma = \frac{1}{2}$

$$\Rightarrow \max_{\gamma,\beta} V_1 = \frac{1}{\sqrt{4e}} \approx 0.303$$

- Can also show **concentration**

$$\lim_{n\to\infty} \mathbb{E}_J[\langle (C/n)^2\rangle] = \lim_{n\to\infty} \mathbb{E}_J^2[\langle C/n\rangle]$$

- Generic vs. optimized QAOA

Suppose $J_{ij} \in \{\pm 1\}$

Then γ is periodic on $[0, \sqrt{n}\pi)$

β is periodic on $[0, \pi/2)$

Generic QAOA state has $\langle C\rangle = e^{-O(n)}$!!

Performance of the QAOA on the SK model

- Higher p : our current method uses $O(4^p)$ memory and $O(16^p)$ time

$$\boxed{V_p(\boldsymbol{\gamma},\boldsymbol{\beta}) = \lim_{n\to\infty}\frac{1}{n}\mathbb{E}_J[\langle\boldsymbol{\gamma},\boldsymbol{\beta}|C|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle]}$$

Performance of the QAOA on the SK model

- Higher p : our current method uses $O(4^p)$ memory and $O(16^p)$ time

$$V_p(\boldsymbol{\gamma},\boldsymbol{\beta}) = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}_J[\langle\boldsymbol{\gamma},\boldsymbol{\beta}|C|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle]$$

p	Best known V_p	
1	0.303265	
2	0.407545	
…	…	
8	0.607266	
$11^\dagger$	0.639311	**beats SDP!**
$12^\dagger$	0.646557	

†unoptimized

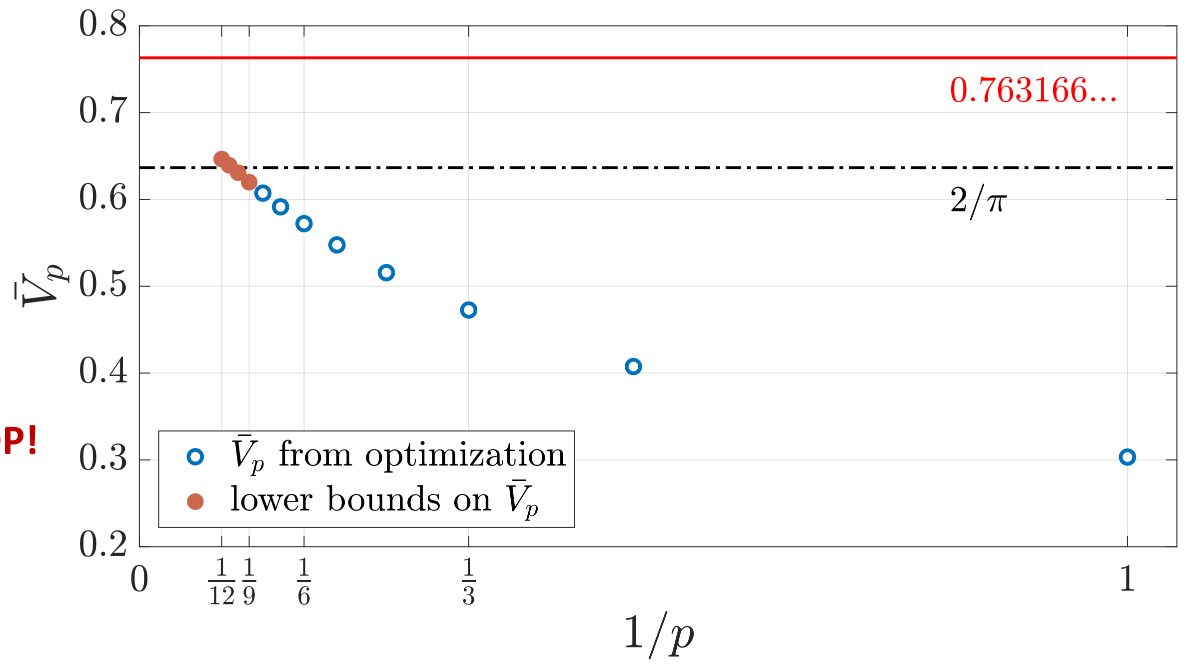
Performance of the QAOA on the SK model

- Higher p : our current method uses $O(4^p)$ memory and $O(16^p)$ time

$$V_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \lim_{n \to \infty} \frac{1}{n} \mathbb{E}_J[\langle \boldsymbol{\gamma}, \boldsymbol{\beta} | C | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle]$$

p	Best known V_p
1	0.303265
2	0.407545
…	…
8	0.607266
$11^\dagger$	0.639311 **beats SDP!**
$12^\dagger$	0.646557

†unoptimized



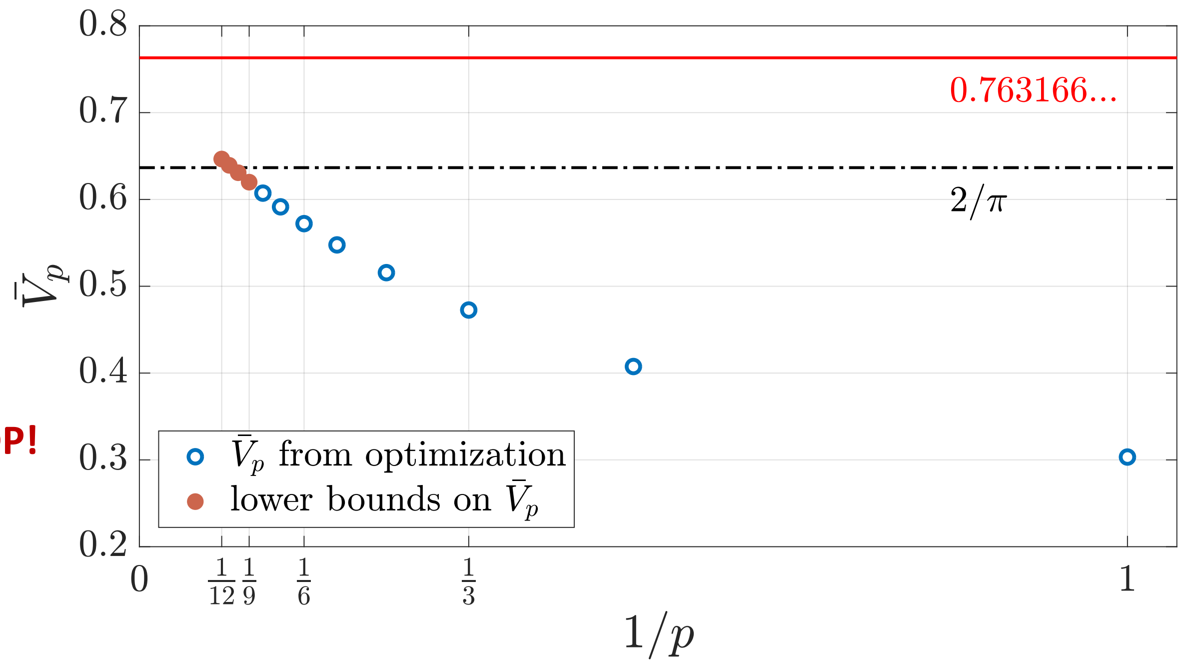
Performance of the QAOA on the SK model

- Higher p : our current method uses $O(4^p)$ memory and $O(16^p)$ time

$$V_p(\boldsymbol{\gamma},\boldsymbol{\beta}) = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}_J[\langle\boldsymbol{\gamma},\boldsymbol{\beta}|C|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle]$$

p	Best known V_p
1	0.303265
2	0.407545
…	…
8	0.607266
$11^\dagger$	0.639311 **beats SDP!**
$12^\dagger$	0.646557

†unoptimized



$$\lim_{p\to\infty}\lim_{n\to\infty} \stackrel{?}{=} \lim_{n\to\infty}\lim_{p\to\infty}$$

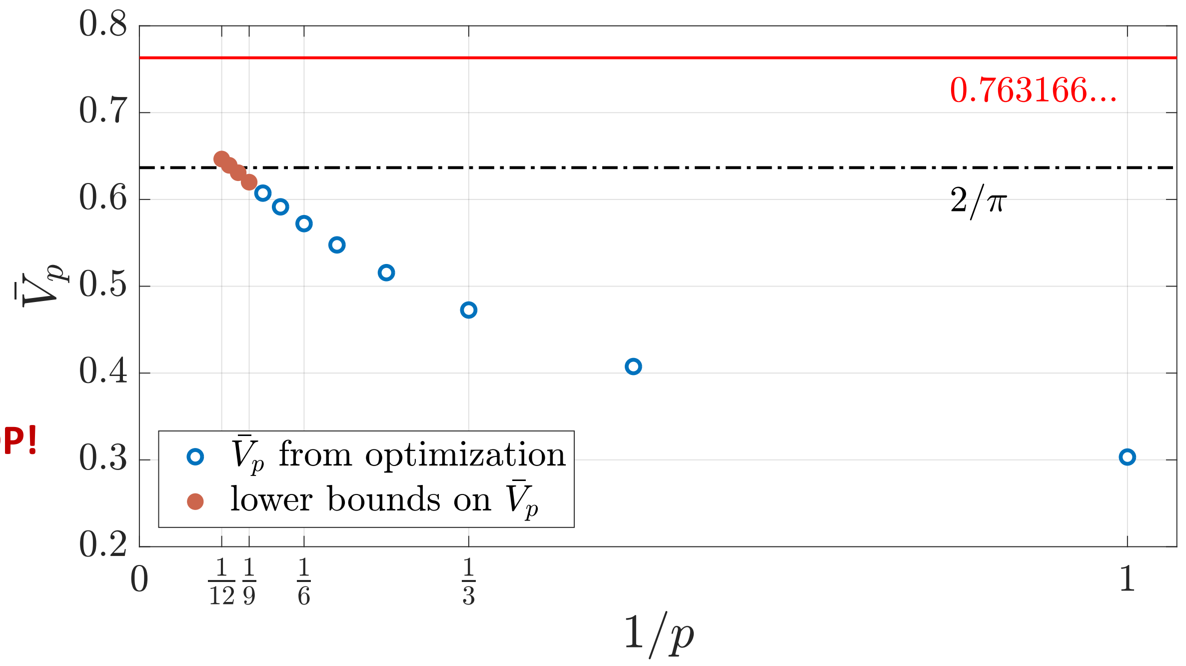
Performance of the QAOA on the SK model

- Higher p : our current method uses $O(4^p)$ memory and $O(16^p)$ time

$$V_p(\boldsymbol{\gamma},\boldsymbol{\beta}) = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}_J[\langle \boldsymbol{\gamma},\boldsymbol{\beta}|C|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle]$$

p	Best known V_p	
1	0.303265	
2	0.407545	
…	…	
8	0.607266	
11†	0.639311	**beats SDP!**
12†	0.646557	

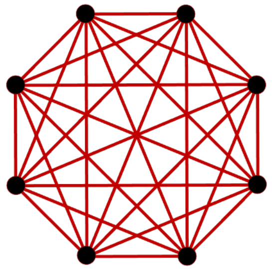
†unoptimized



If $\lim_{p\to\infty} \bar{V}_p = \Pi_*$, then a power law fit of optimized V_p yields

$$\bar{V}_p \approx \Pi_* - \frac{1.2}{(p+2)^{0.9}}$$

Summary



- We *analytically* obtain a formula for **typical case performance** of the QAOA on the SK model at high p
  - Evaluation takes $O(16^p)$ currently but may be improvable

- QAOA **beats** Semi-Definite Programming at $p = 11$

- **Concentration** over instances and measurements

[Farhi Goldstone Gutmann **LZ**, arXiv:1910.08187]

https://github.com/leologist/QAOA-SK

Outlook

- Show **convergence** of QAOA as $p \to \infty$? $\lim_{p\to\infty} \lim_{n\to\infty} \stackrel{?}{=} \lim_{n\to\infty} \lim_{p\to\infty}$

Outlook

- Show **convergence** of QAOA as $p \to \infty$? $\lim_{p\to\infty} \lim_{n\to\infty} \stackrel{?}{=} \lim_{n\to\infty} \lim_{p\to\infty}$

- **Average over instances** for harder problems for provable speedup?

q-spin model

$$C = \sum_{i_1 < \cdots < i_q} J_{i_1 \ldots i_q} Z_{i_1} \cdots Z_{i_q}$$

Provably hard for classical algorithms due to their "**Overlap Gap Property**"

[Gamarnik Jagannath 2019]
[Gamarnik Jagannath Wein 2020]

Montanari's algorithm *stuck at 98.4% approximation ratio for q=3*
[Alaoui Montanari 2020]

QAOA @ p=1
gets 33% for q=3