On Benchmarking Quantum Heuristics

Zhihui Wang

zhihui.wang@nasa.gov

Quantum AI Lab, NASA Ames Center
Universities Space Research Association

May 02 2019

IEEE Quantum Conference

Benchmarking quantum algorithms

- Quantum speedup:
  - Any possible classical algorithms — Proven speedup, Shor, Grover
  - State-of-art classical algorithms
  - A general-purpose classical algorithm — E.g., SA, QMC

[Rønnow, Wang, Job et. al., *Defining and detecting quantum speedup*, Science 2014]

- A different type of algorithms: quantum heuristics
  - Quantum annealing (QA)
  - Variational quantum eigen solver (VQE)
  - Quantum approximate optimization algorithm (QAOA)

suitable for NISQ device

Purpose:

No guaranteed speedup as a general algorithm

Approximate optimization

Exact optimization

Sampling: specific distribution, fair sampling

Universal QC

Universality proven through demonstrating ability to generate universal basis sets.

[Lloyd, Quantum approximate optimization is computationally universal, arXiv:1812.11075]

Benchmarking quantum heuristics

Aspects

- Purpose:
  - — Approximate optimization
  - — Exact optimization
  - — Sampling: specific distribution, fair sampling
  - — Universal QC

Scaling with *n*

- Metrics:
  - — approximation ratio
  - — prob-to-exact-solution,
  - — fairness, distance between distributions
  - — quantum complexity

- Additional Metrics for NISQ
  - — Circuit depth
    - All-to-all connectivity of physical qubits
    - Easier to scale connectivity (2D grid, Google bristlecone, Rigetti, IBM)
    - Logical quantum circuit to physical circuit: Circuit Compilation (gate scheduling)
    - Choice of basis gate sets
  - — Robustness
    - Variational nature will tolerate certain errors.
    - More actively sought fault-tolerance?
  - — Classical parameter setting
    - Analytical methods: deterministic parameters
    - How to update parameter values (gradient based, statistical optimization)
    - Quantum control landscape: local minima; barren plateau
  - — Benchmarking problem set
    - Typical vs worst-case
    - Small and hard

Illustrating using QAOA for graph coloring

- — Choice of Phase-Separator and Mixer
  - Use cost function (standard)
  - Specially designed: 1D chain with parity-dependent parameters (universal QC)
- — Choice of initial states
- — Measurement: What can we infer from the expected value / average performance?

— Circuit depth
  - All-to-all connectivity of physical qubits
  - Easier to scale connectivity (2D grid, Google bristlecone, Rigetti, IBM)
  - Logical quantum circuit to physical circuit: Circuit Compilation (gate scheduling)
  - Choice of basis gate sets

— Robustness
  - Variational nature will tolerate certain errors.
  - More actively sought fault-tolerance?

— Classical parameter setting
  - Analytical methods: deterministic parameters
  - Variational:
    - How to update parameter values (gradient based, statistical optimization)
  - Quantum control landscape: local minima; barren plateau

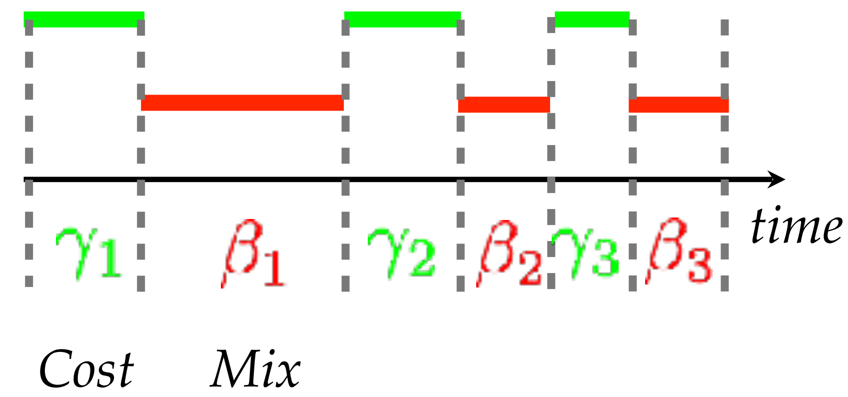
— Benchmarking problem set
  - Typical vs worst-case
  - Small and hard

One leading Candidate of quantum heuristics: Quantum Approximate Optimization Algorithms (QAOA) —> Quantum Alternating Operator Ansatz

- One-line summary of the algorithm

$$U = e^{-i\beta_p H_M} e^{-i\gamma_p H_C} \cdots e^{-i\beta_2 H_M} e^{-i\gamma_1 H_C}$$

[Farhi, Goldstone, and Gutmann, arXiv:1411.4028]

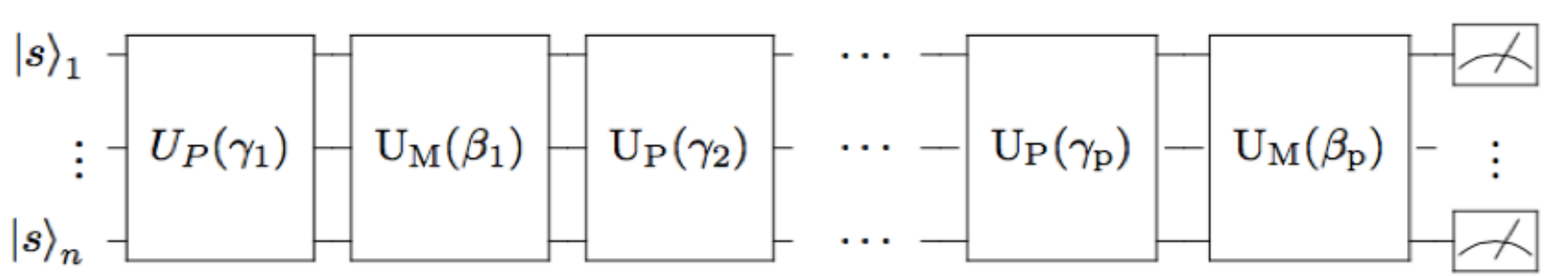


$$H_C = \sum_{j,j',\cdots} (C_j\sigma_j^z + C_{j,j'}\sigma_j^z\sigma_{j'}^z + C_{j,j',j''}\sigma_j^z\sigma_{j'}^z\sigma_{j''}^z, + \cdots)$$

$$H_M = \sum_j \sigma_j^x$$

Approach: $QAOA_p$ circuit:

- Prepare initial state
- Loop p times, and on iteration i apply Hamiltonians
  - (Phase separation) Cost-function-based H_C diag. in Z basis, for time γ_i
  - (Mixing) Hamiltonian H_M, for time β_i
- Measure in computational basis



Choice of Mixer: QAOA for constrained optimization

- How are constrained problems approached?

Encode the constraints as penalty in the cost function. — Lagrange multipliers

Commonly practiced in quantum annealing.

Alternative: Use a mixer/driver that contains the quantum evolution in the subspace that satisfies the constraints.

Original motivation: Alleviate embedding burden **[Hen & Spedalieri, 2016]**

Another Advantage: Smaller search space!

We extend this idea to QAOA, formulate such mixers for a number of problems

Concept: [Hadfield, Wang, O'Gorman, Rieffel, Venturelli, Biswas, arXiv 1709.03489];

and **Performance & Circuit: [Wang, Rubin, Dominy, Rieffel, arXiv:1904.09314]**

study the performance of such alternate mixers

QAOA for graph coloring problem

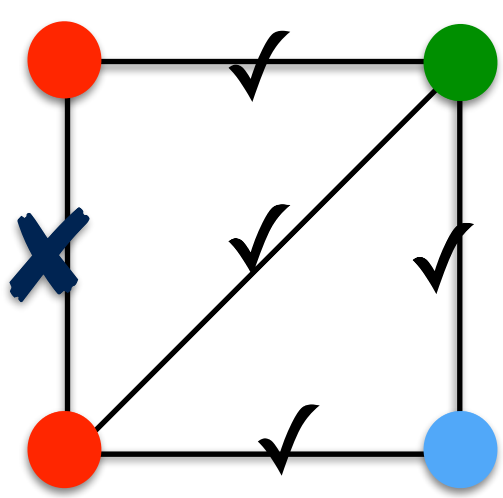
- Goal: Assign colors to vertices to maximize properly-colored edges (connecting two vertices of different color)

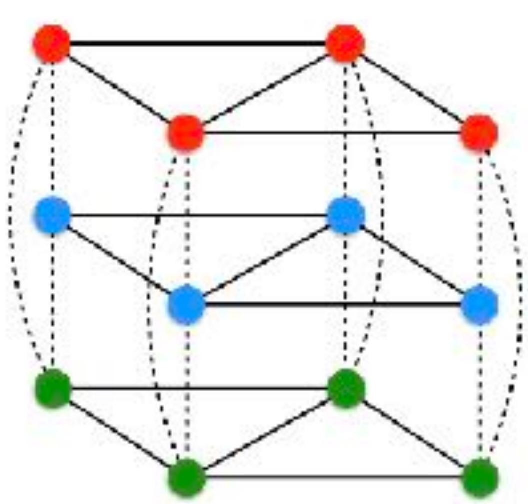
Encoding:

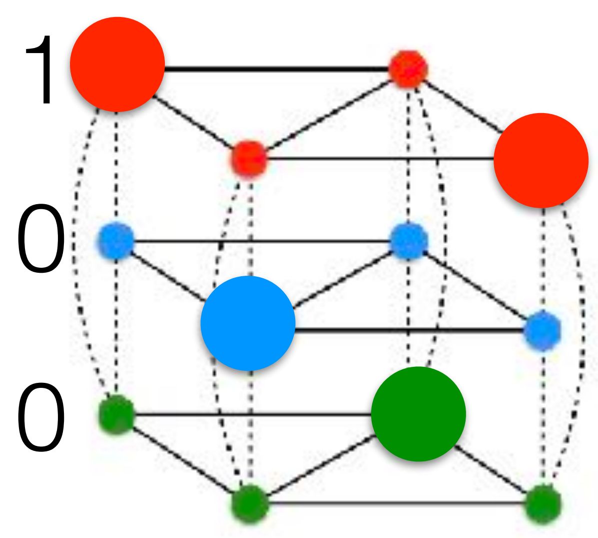
Binary: $x_{v,c} = 1$ whether vertex v is assigned color-c

Constraints:

Each vertex should have exactly one color: $\sum_{c=1}^{k} x_{v,c} = 1 \Longleftrightarrow \sum_{c=1}^{k} \sigma^z_{v,c} = k-2$







QAOA for graph coloring problem

Constraints:

Each vertex should have exactly one color: $\sum_{c=1}^{k} x_{v,c} = 1 \Longleftrightarrow \sum_{c=1}^{k} \sigma^z_{v,c} = k-2$

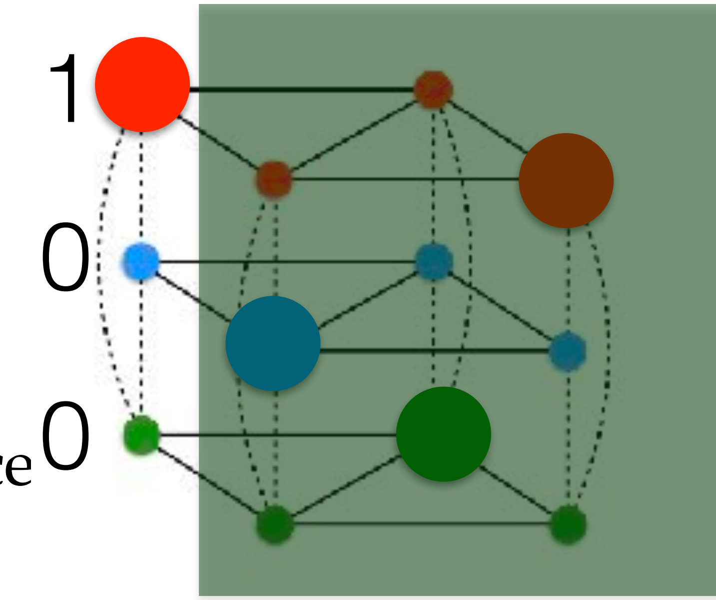
Implemented as penalty in cost: $H_{\texttt{penalty}} = \left[\sum_{c=1}^{k} \sigma^z_{v,c} - (k-2)\right]^2$

Or

Stay in the feasible subspace: $\sum_{c=1}^{k} \sigma^z_{v,c} = k-2$

XY-model $|\uparrow\downarrow\rangle\langle\downarrow\uparrow| + |\downarrow\uparrow\rangle\langle\uparrow\downarrow|$

$\propto \sigma^x_{v,c}\sigma^x_{v,c} + \sigma^y_{v,c}\sigma^y_{v,c}$



Advantage:

Smaller search space: evolution contained in feasible subspace

Closer to hardware:

XY interaction (or iswap-gate) can naturally happen on certain solid-state QC candidate systems

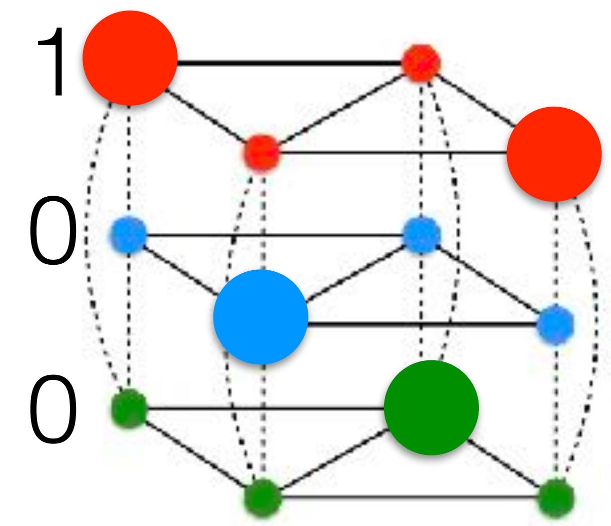
QAOA for graph coloring problem

Cost function

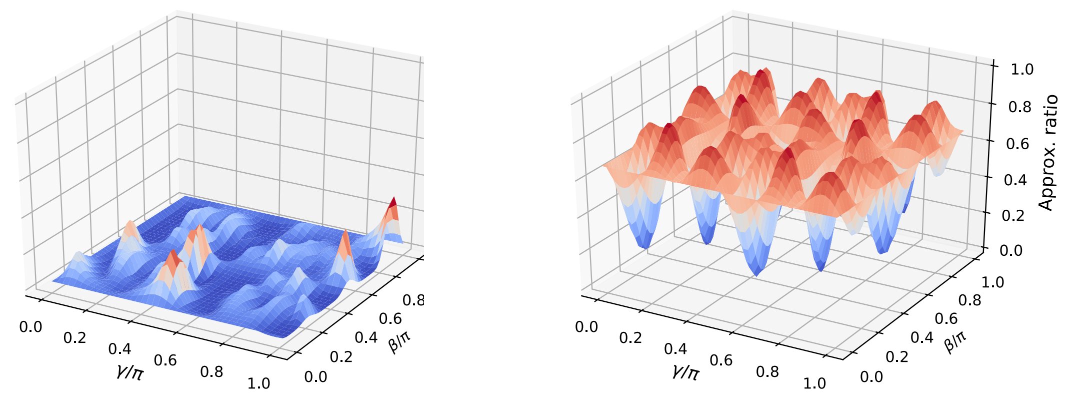
$$f_C = m - \sum_{c=1}^{k} \sum_{\{v,v'\}\in E} x_{v,c} x_{v',c}$$

Cost Hamiltonian

$$H_C = \sum_{c=1}^{k} \sum_{\{v,v'\}\in E} \sigma^z_{v,c} \sigma^z_{v',c}$$

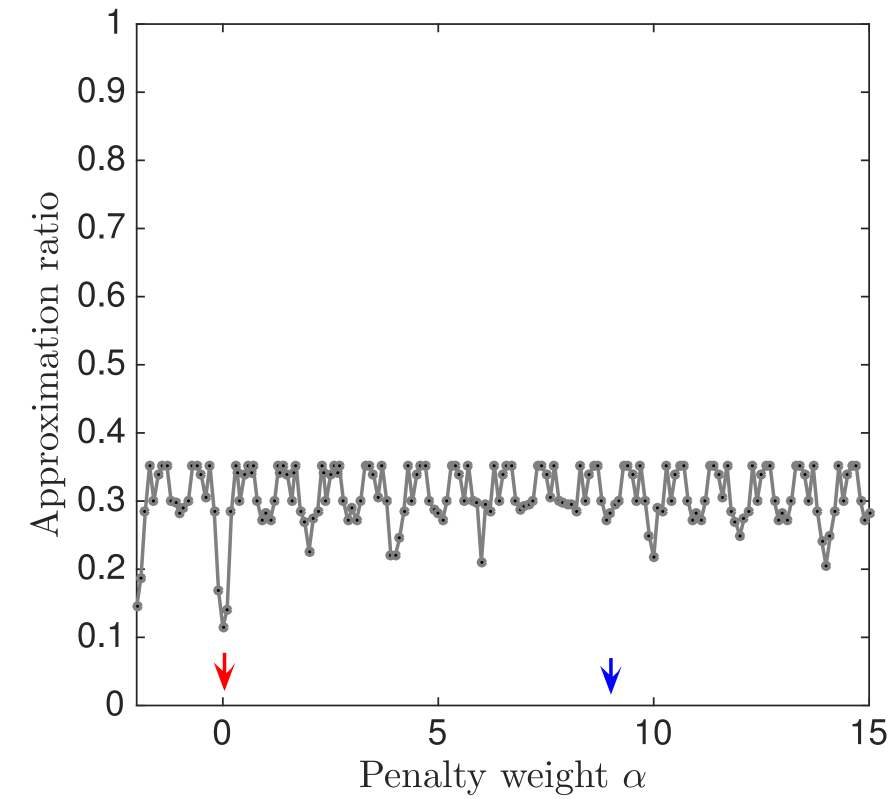


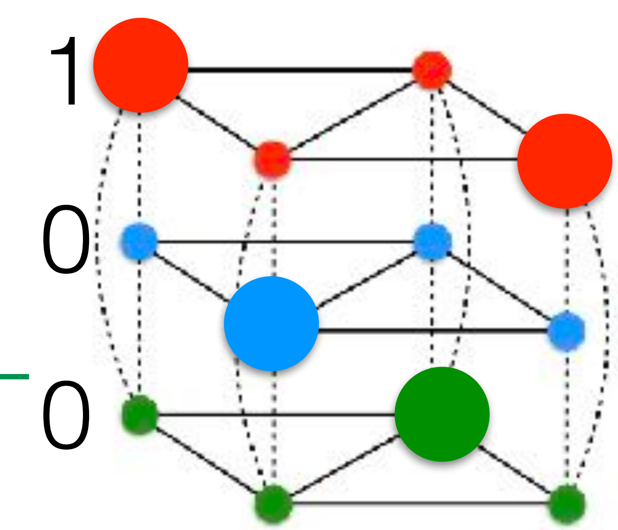
Choice of “Cost Hamiltonian”: Role of energy in QAOA?



(a) (b)

Figure 3. Numerical results for level 1 $QAOA$ on the problem of 3-coloring of a triangle graph. (a) using X mixer along with phase-separating Hamiltonian, Eq. (8) where the penalty weight is taken to be the numerically determined optimal value $\alpha^* = 1.7$. (b) using the XY mixer with W-state being the initial state.





- Size of search space

Penalty + X-mixer — Full Hilbert space — 2^{nk}

XY mixer — Feasible subspace — k^n

Ratio: $\left(\frac{k}{2^k}\right)^n$ — The feasible space **shrinks exponentially** with *n*.

Bench marking problem sets: What graphs to color on NISQ era hardware?

Small & Hard graphs

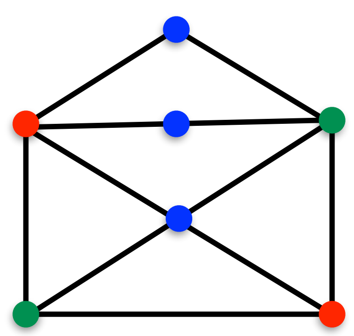
- For a classical algorithm, there is a concept of

smallest slightly-hard-to-color graph: *applying the algorithm will **sometimes** yield the optimal solution*

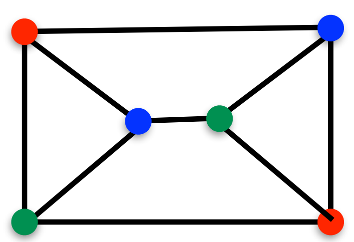
&

smallest hard-to-color graph: *applying the algorithm **never** yields the optimal solution*

- Examples



Envelope



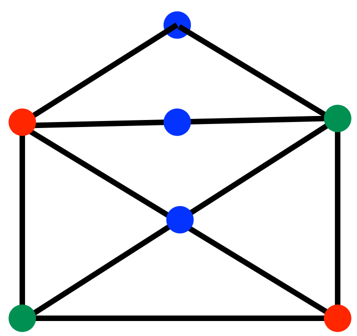
Prism

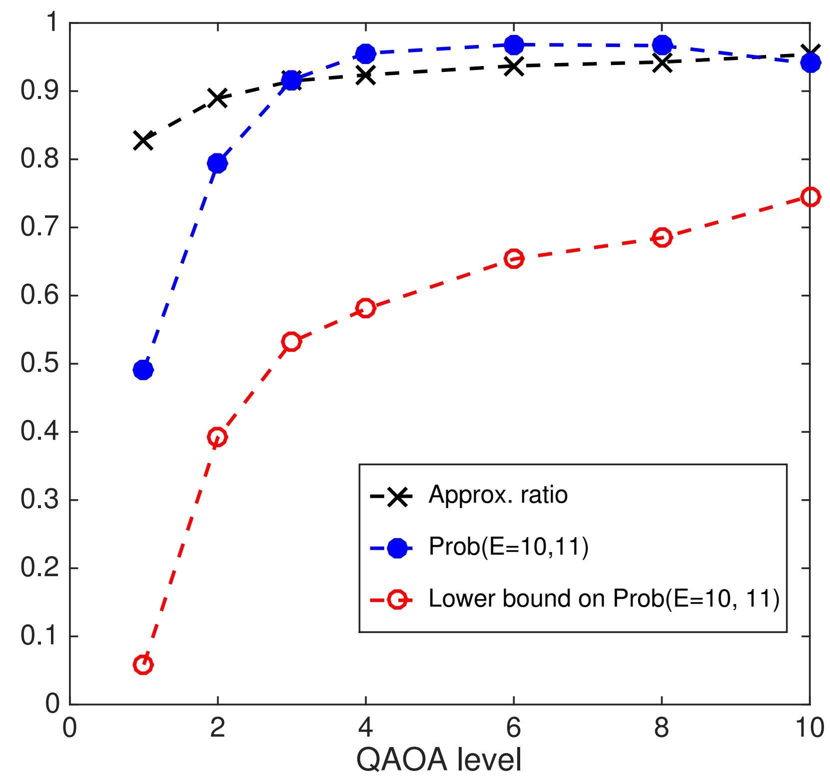
Measurement/Parameter update: What can we infer from the expected value / average performance?

$X \in \{0, 1, \ldots, m\}$, if the mean value is μ then for any $l \leq \lfloor \mu \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function, the probability of x taking value larger than l is lower-bounded as

$$\Pr(X > l) \geq \frac{\mu - l}{m - l} \, . \tag{1}$$

For combinatorial optimization: High mean often accompanies high typical value





Choice of initial states

Generalized W-state: For any number of qubits
superposition of classical states of Hamming-weight 1.

Eigen-state of the XY mixer, an uniform superposition of all feasible classical states

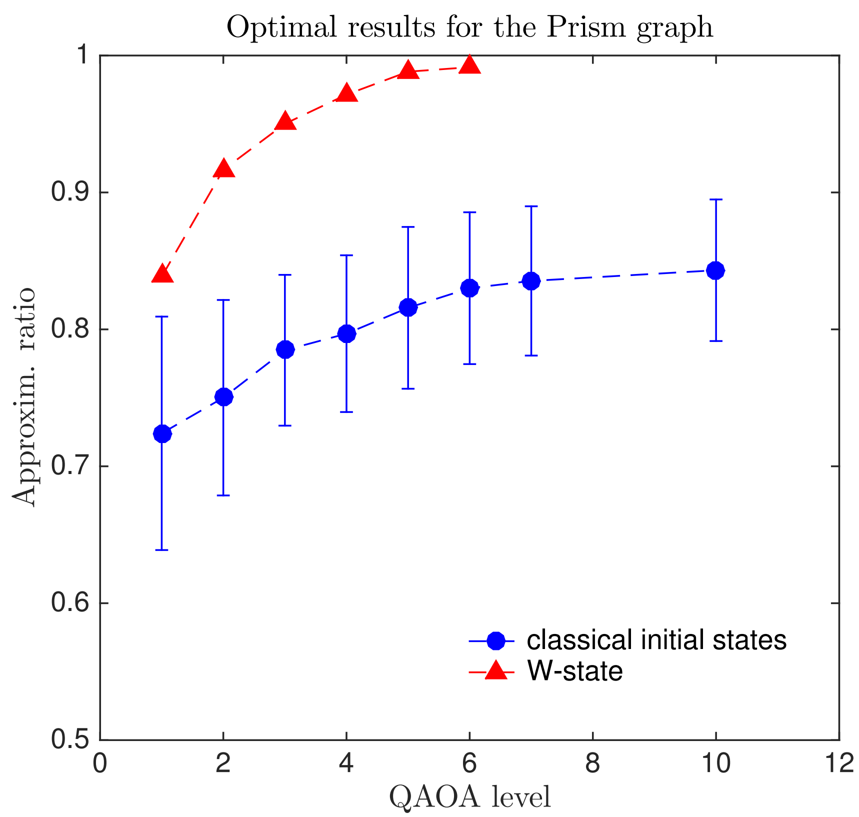
$$|W\rangle_v = \frac{1}{\sqrt{k}}(|100\cdots0\rangle + |010\cdots0\rangle + \cdots + |0\cdots01\rangle)$$

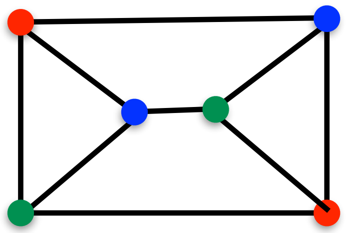
Classical initial states: random-coloring of the graph

$$|\psi_0\rangle = |100\cdots0\rangle \otimes |010\cdots0\rangle \otimes \cdots \otimes |0\cdots01\rangle$$

Easier to prepare:
n single-qubit gates

Classical initial state: easy to generate
vs
W-state: Better performance



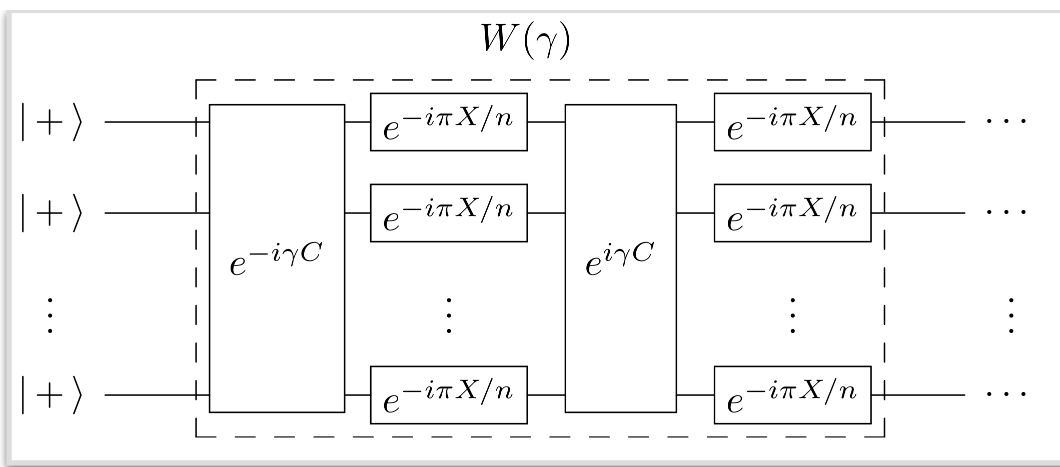


Parameter setting

QAOA for Grover's problem

Deterministic parameters:
beta=pi/n, gamma=arbitrary

$$W(\gamma) = e^{-i\pi B/n} e^{i\gamma C} e^{-i\pi B/n} e^{-i\gamma C}$$



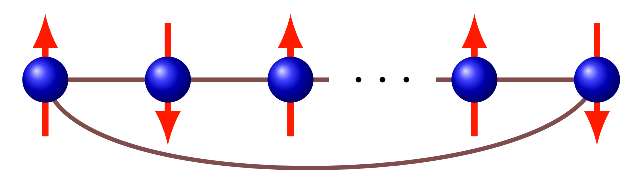
Reproduces the √N quantum speedup

$$\langle \boldsymbol{u} \,|\, \psi_{\text{out}} \rangle \simeq 1/\sqrt{2}\,, \qquad p \simeq \frac{\pi}{4\sqrt{2}}\sqrt{N}$$

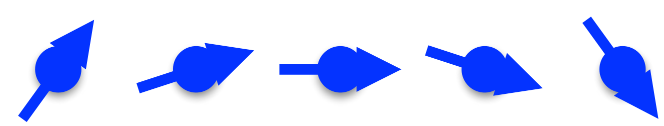
[Jiang, Rieffel, Wang, Near-optimal quantum circuit for Grover's unstructured search using a transverse field, PRA 2017]

QAOA for AF ring (MaxCut on a ring)

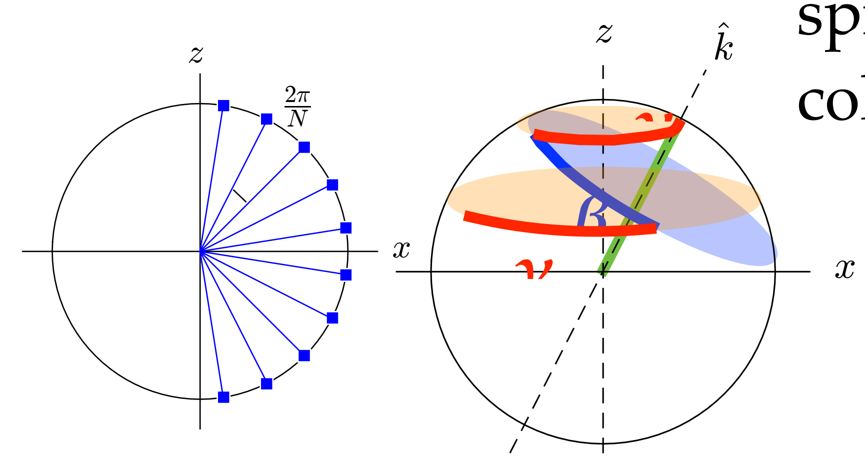
- Anti-Ferromagnetic Chain: $H_C = \sum_j \sigma_j^z \sigma_{j+1}^z$
- Analysis: Jordan-Wigner transformation for 1D spin chain with n.n. couplings



coupled spin chain of size N

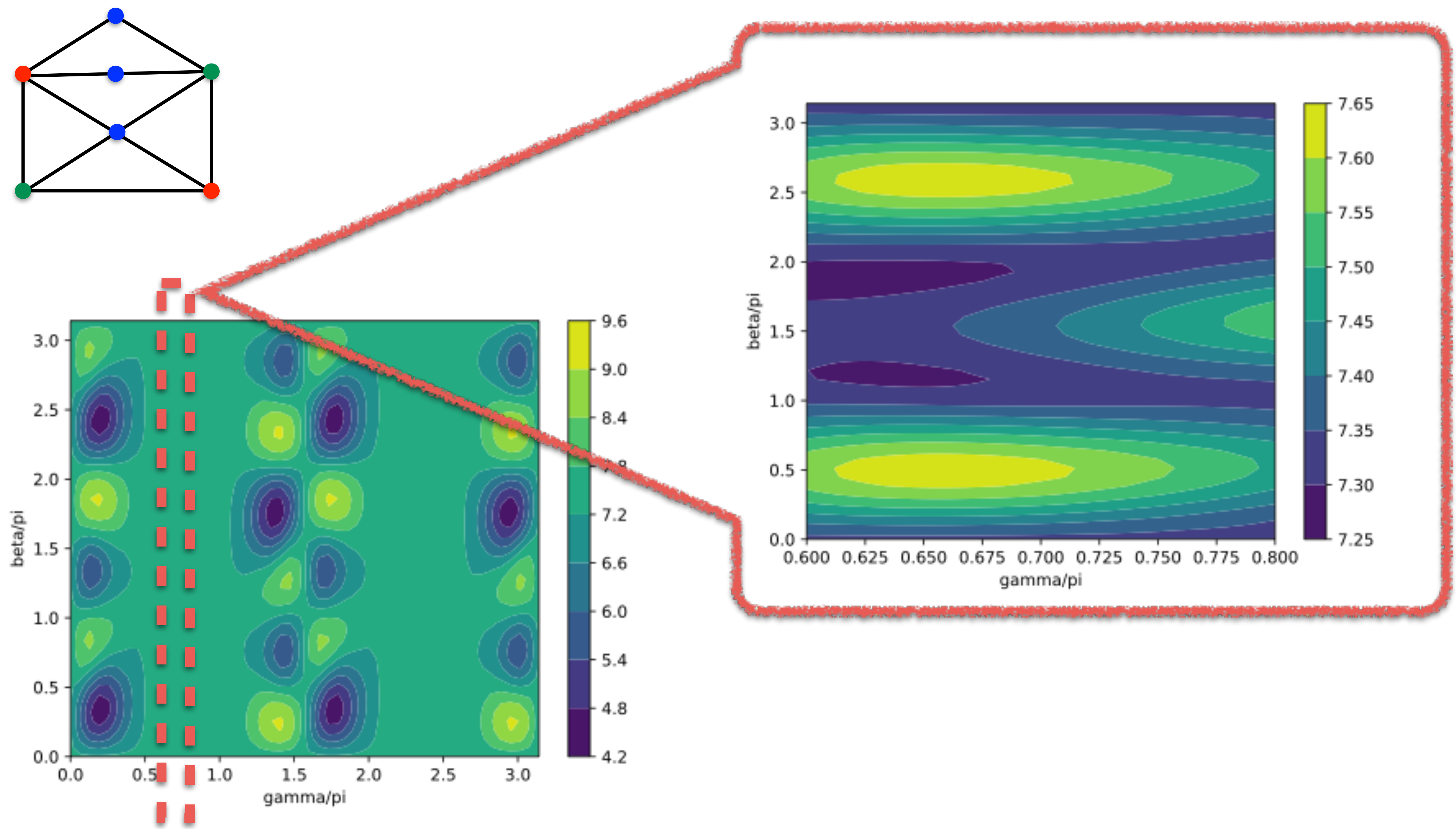


2-axis control of N/2 independent spins with collective angles



[Wang, Hadfield, Jiang, Rieffel, QAOA for MaxCut: a fermonic review, PRA 2018]

Quantum control landscape: local minima; barren plateau



Rugged landscape — stochastic optimizing is needed

Summary

- We outlined important aspects of benchmarking quantum heuristics
- Using QAOA with XY mixer as an example, we demonstrated that influences to algorithm performance could come from
  - Design principle
    — Choice of "Cost function": challenges the guidance role of energy in QAOA
    — Choice of Mixers: contains search in feasible subspace satisfying constraints
    — Choice of initial state: tradeoff between good (noise-free) performance and complexity of state-preparation
  - Implementation on hardware
  - — Circuit-depth for XY gates: can be efficiently implemented on hardware: from all-to-all to a chain connectivity
  - Parameter setting and Quantum control landscape

XY mixers for QAOA: [Wang, Rubin, Dominy, Rieffel, *arXiv*:1904.09314]

QAOA for Grover: [Jiang, Rieffel, Wang, *PRA* 2017]

From QAOA to QAOA: [Hadfield, Wang, O'Gorman, Rieffel, Venturelli, Biswas, *Algorithms* 2019]

QAOA for MaxCut: a fermonic view,[Wang, Hadfield, Jiang, Rieffel, *PRA* 2018]


Thank You!
NASA Advanced Supercomputing Facility
N258