A Brief Introduction to the QAOA

Andreas Bärtschi
CCS-3 Information Sciences

CPAIOR 2024

LA-UR-24-25452

Quantum Approximate Optimization Algorithm (QAOA) / Quantum Alternating Operator Ansatz (QAOA+)

1. A heuristic to solve unconstrained / constrained combinatorial optimization problems
2. Either approximately (metric: approximation ratio) or exactly (metric: probability of sampling optimal solution)

Agenda

01

Quantum Approximate Optimization Algorithm

- Quantum subroutine
- Cost Hamiltonian
- Optimization Goals
- Parameter Finding

02

Quantum Alternating Operator Ansatz

- Quantum subroutine
- State Preparation
- Mixer Design
- Conclusion and Summary

03

Deep Dive: Hamming weight constraint

Quantum Approximate Optimization Algorithm

(QAOA)

Quantum Approximate Optimization Algorithm

Farhi, Goldstone, Gutmann (arXiv:1411.4028)

The Quantum Approximate Optimization Algorithm (QAOA)
is a heuristic for unconstrained combinatorial optimization. It prepares a state

$$|\boldsymbol{\beta},\boldsymbol{\gamma}\rangle = e^{-i\beta_p \Sigma X_i} e^{-i\gamma_p H_C} \dots e^{-i\beta_1 \Sigma X_i}\, e^{-i\gamma_1 H_C} |+^n\rangle$$

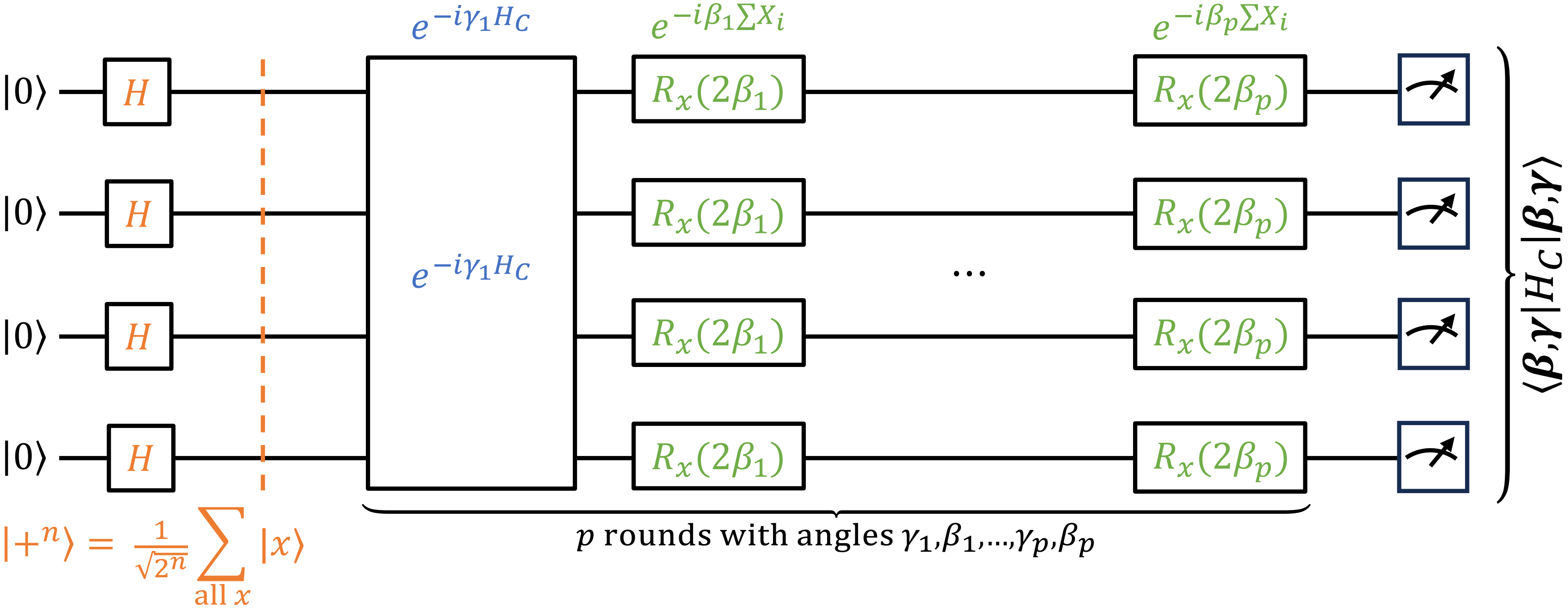
from which one would like to sample good solutions with high probability.

QAOA is specified by

- the initial state $|+^n\rangle$, the uniform superposition over all basis states,
- a Cost Hamiltonian H_C,
  used to phase basis states proportional to their cost $C(x)$: $|x\rangle \mapsto e^{-i\gamma C(x)}|x\rangle$
  and to evaluate the expectation value of the QAOA state, $\langle \boldsymbol{\beta},\boldsymbol{\gamma}|H_C|\boldsymbol{\beta},\boldsymbol{\gamma}\rangle$,
- a Transverse Field Mixer $\sum_{i=1}^{n} X_i$,
  used to interfere states in order to boost good solutions,
- p rounds with individual angle parameters $\beta_1, \dots, \beta_p, \gamma_1, \dots, \gamma_p$.

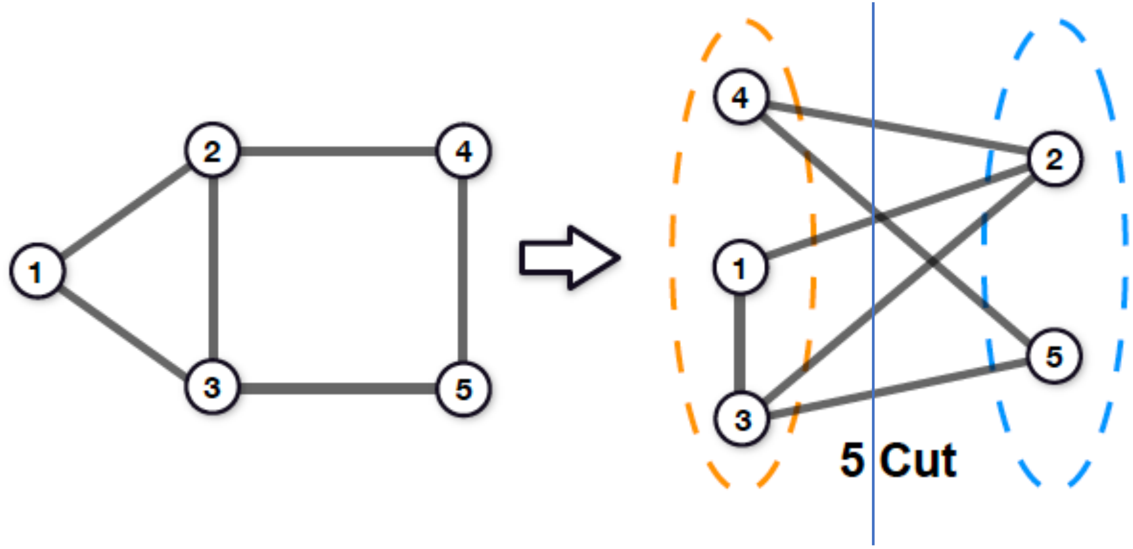
QAOA quantum subroutine / circuit

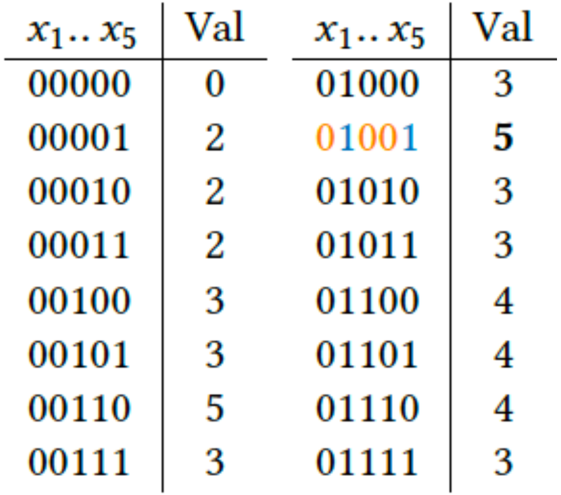
Sample solutions from a QAOA circuit with classically determined parameters:


$e^{-i\gamma_1 H_C}$
$e^{-i\beta_1 \Sigma X_i}$
$e^{-i\beta_p \Sigma X_i}$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
H
H
H
H
$e^{-i\gamma_1 H_C}$
$R_x(2\beta_1)$
$R_x(2\beta_1)$
$R_x(2\beta_1)$
$R_x(2\beta_1)$
$\cdots$
$R_x(2\beta_p)$
$R_x(2\beta_p)$
$R_x(2\beta_p)$
$R_x(2\beta_p)$
$\langle \boldsymbol{\beta}, \boldsymbol{\gamma} | H_C | \boldsymbol{\beta}, \boldsymbol{\gamma} \rangle$
$|+^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{\text{all } x} |x\rangle$
p rounds with angles $\gamma_1, \beta_1, \ldots, \gamma_p, \beta_p$

Cost Hamiltonian: Construction

MaxCUT on $G = (V, E),\ n = |V|,\ m = |E|$





$x_1..x_5$	Val	$x_1..x_5$	Val
00000	0	01000	3
00001	2	01001	5
00010	2	01010	3
00011	2	01011	3
00100	3	01100	4
00101	3	01101	4
00110	5	01110	4
00111	3	01111	3

Quantum Algorithm Implementations for Beginners, arXiv:1804.03719

Cost Hamiltonian: Construction

MaxCUT on $G = (V, E)$, $n = |V|$, $m = |E|$

1. Start with Cost function $C(x)$ on states x given by n binary variables $x \in \{0,1\}^n$

$$C(x): \sum_{\{u,v\} \in E} x_u \oplus x_v = \sum_{\{u,v\} \in E} x_u + x_v - 2x_u x_v$$

2. Transform to spin variables $s \in \{1, -1\}^n$ using $x_i = (1 - s_i)/2$

$$C(s): \sum_{\{u,v\} \in E} \frac{1 - s_u s_v}{2} = \frac{m}{2} - \sum_{\{u,v\} \in E} \frac{s_u s_v}{2}$$

3. Replace terms with qubit operators: spin variables by Pauli-Z, constants by Identities I

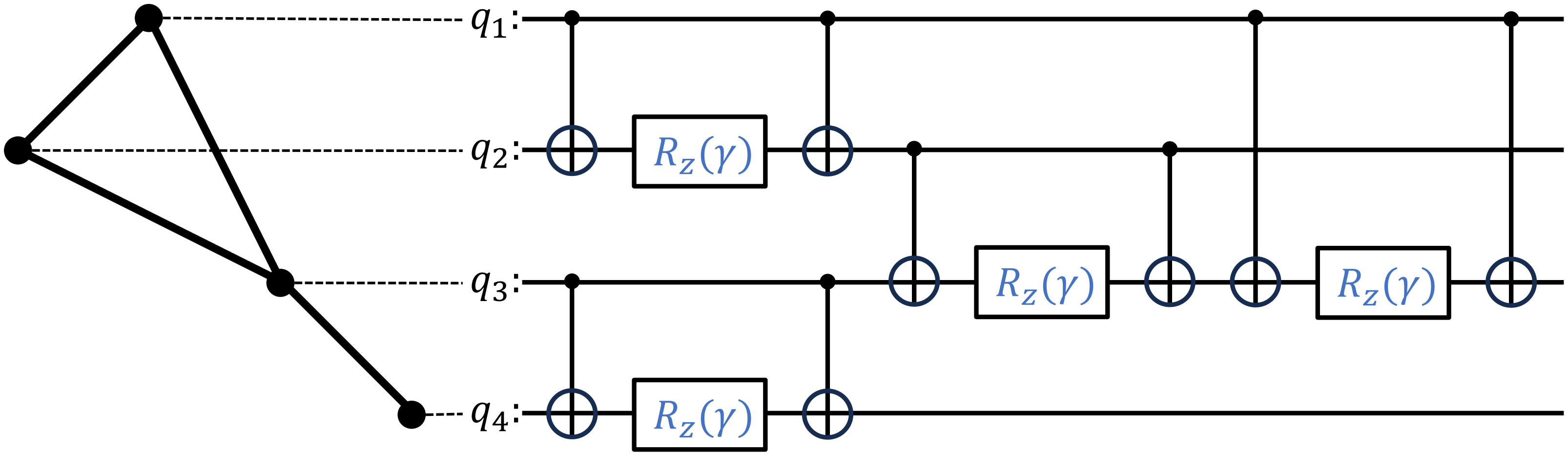
$$H_C: \sum_{\{u,v\} \in E} \frac{I - Z_u Z_v}{2} = \frac{m}{2} I - \sum_{\{u,v\} \in E} \frac{Z_u Z_v}{2}$$

Quantum Algorithm Implementations for Beginners, arXiv:1804.03719

Cost Hamiltonian: Implementation

MaxCUT on $G = (V, E),\ n = |V|,\ m = |E|$

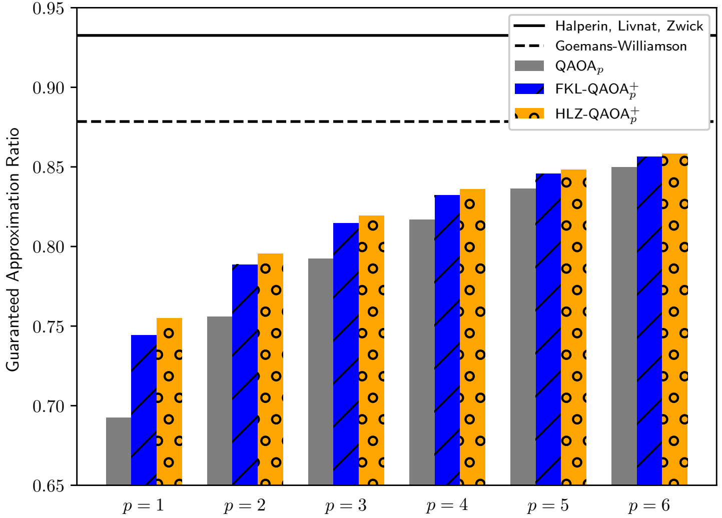
Implement $e^{-i\gamma H_C}$ term by term: $H_C = \sum_{\{u,v\}\in E} \frac{I - Z_u Z_v}{2}$

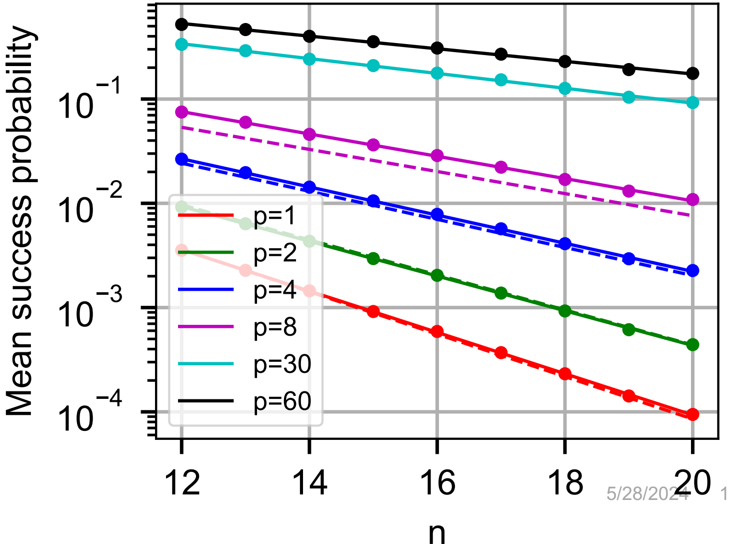


Optimization Goals

- Approximate Optimization
  Provable or numerically observed good **approximation ratios** from the expectation value $\langle \boldsymbol{\beta},\boldsymbol{\gamma}|H_C|\boldsymbol{\beta},\boldsymbol{\gamma}\rangle / \max_x C(x)$
  e.g. for MaxCut on large-girth 3-regular graphs (Caha, Kliesch, Koenig, arXiv:2203.00717)

- Exact Optimization
  Provable or numerically observed "large" overlap (success probability) with an optimum solution e.g. for finding a satisfiable assignment to satisfiable E8-SAT formula of increasing sizes (Boulebnane, Montanaro, arXiv:2208.06909)





Parameter Finding: Methods and Challenges

- Classical optimizer in an outer loop with quantum part as a subroutine: gradient descent, basin hopping, derivative-free methods
  Challenge: barren plateaus, rugged parameter landscapes
- Parameter transfer from smaller instances
  Challenge: saturation at low depth QAOAs on small instances
- Interpolation from smaller number of rounds
  Challenge: local minima, overhead
- Linear ramp, discretizations of adiabatic schedules
  Challenge: requires large number of QAOA rounds
- Fixed angles approaches
  Challenge: not generalizable to other problems

Quantum Alternating Operator Ansatz

(QAOA+)

Quantum Alternating Operator Ansatz

Hadfield, Wang, O’Gorman, Rieffel, Venturelli, Biswas (arXiv:1709.03489)

The Quantum Alternating Operator Ansatz (QAOA+)
is a heuristic for combinatorial optimization. It prepares a state

$$|\boldsymbol{\beta},\boldsymbol{\gamma}\rangle = U_M(\beta_p)U_P(\gamma_p)\cdots U_M(\beta_1)U_P(\gamma_1)U_S|0^n\rangle$$

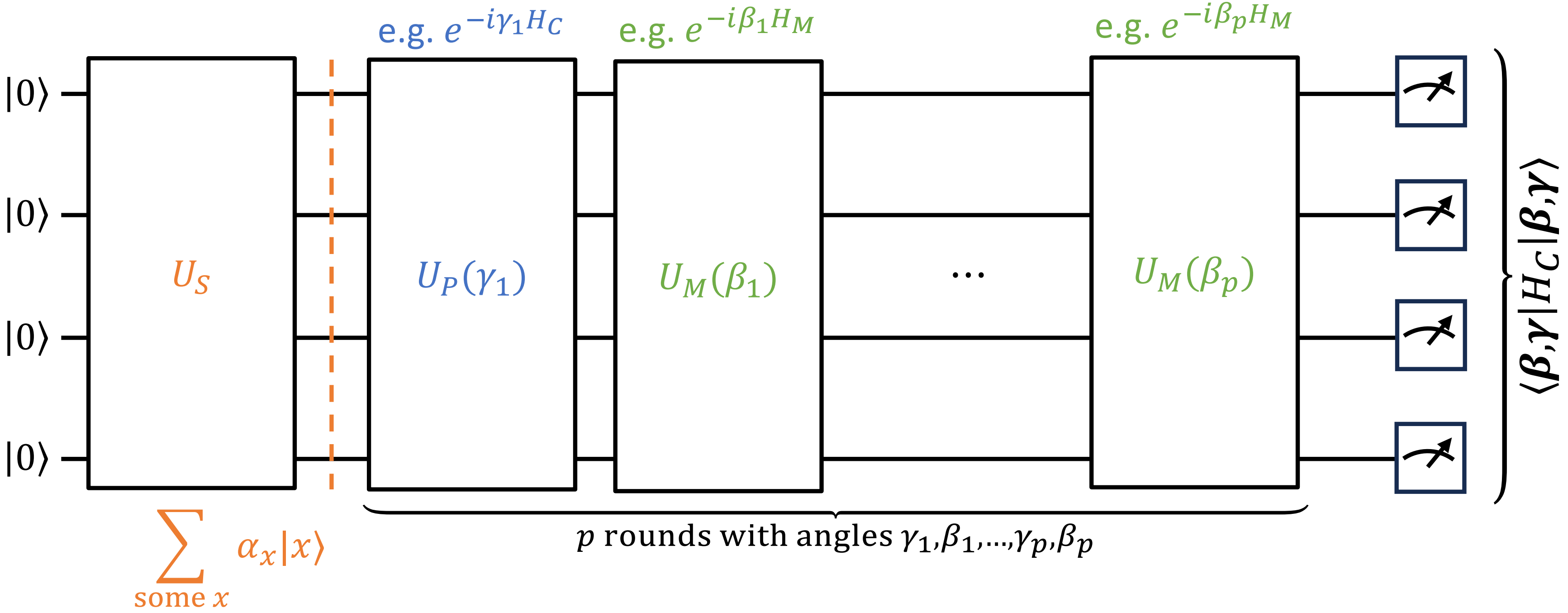
from which one would like to sample good solutions with high probability.

QAOA+ is specified by

- an initial state preparation unitary U_S,
  preparing some superposition of feasible states.
- a phase separator unitary $U_P(\gamma)$,
  used to phase basis states by some function f of their cost $C(x)$: $|x\rangle \mapsto e^{-i\gamma f(C(x))}|x\rangle$,
- a mixer unitary $U_M(\beta)$, preserving and interfering feasible solutions,
  e.g., $e^{-i\beta H_M}$ with H_M adjacency matrix with connected component of all feasible states,
- p rounds with individual angle parameters $\beta_1, \dots, \beta_p, \gamma_1, \dots, \gamma_p$.

QAOA+ quantum subroutine / circuit

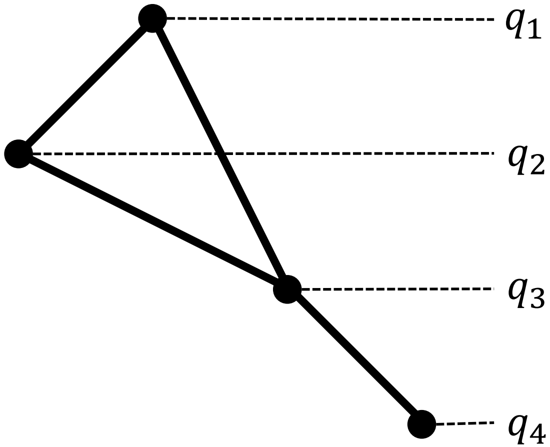
Sample solutions from a QAOA circuit with classically determined parameters:


e.g. $e^{-i\gamma_1 H_C}$
e.g. $e^{-i\beta_1 H_M}$
e.g. $e^{-i\beta_p H_M}$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
U_S
$U_P(\gamma_1)$
$U_M(\beta_1)$
$\cdots$
$U_M(\beta_p)$
$\langle \boldsymbol{\beta},\boldsymbol{\gamma} | H_C | \boldsymbol{\beta},\boldsymbol{\gamma} \rangle$
$\sum_{\text{some } x} \alpha_x |x\rangle$
p rounds with angles $\gamma_1, \beta_1, \ldots, \gamma_p, \beta_p$

State Preparation for k-Densest Subgraph

Maximize number of edges in a subgraph induced by k vertices (here $k = 3$, $n = 4$).

Feasible states are given by a Hamming weight constraint



Solution Values

3: $|0111\rangle$

2: $|1110\rangle$

$|1101\rangle$

1: $|1011\rangle$

Initial State

U_S: Prepare superposition of all 4 feasible states:

$$|D_k^n\rangle = \frac{1}{\sqrt{\binom{n}{k}}} \sum_{\substack{x\in\{0,1\}^n \\ |x|=k}} |x\rangle$$

$|D_k^n\rangle$ efficiently preparable for arbitrary k (arXiv:2207.09998)

Caveat: Uniform superpositions of feasible states are often beneficial (arXiv:2006.00354) but can not always be implemented efficiently (e.g. for Maximum Independent Set).

Mixer Design for k-Densest Subgraph

Mixers must preserve and provide transitions in the feasible Hamming-Weight k subspace. For example: Mixer Hamiltonian H_M that is an adjacency matrix of a graph on all states, in which the feasible states form a connected component:

$$\boldsymbol{U_M(\beta) = e^{-i\beta\, H_M}}$$

- **Ring Mixer** $H_M = \frac{1}{2}\sum_{j=i+1} X_i X_j + Y_i Y_j$
- **Clique Mixer** $H_M = \frac{1}{2}\sum_{j>i} X_i X_j + Y_i Y_j$
- **Grover Mixer** $H_M = |D_k^n\rangle\langle D_k^n|$

$$H_{XY} = \frac{1}{2}(X_1 X_2 + Y_1 Y_2) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$e^{-i\beta/2(X_1X_2+Y_1Y_2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta/2 & -i\sin\beta/2 & 0 \\ 0 & -i\sin\beta/2 & \cos\beta/2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

More ad-hoc mixer unitaries, not necessarily based on Hamiltonians, are also possible, as are Hamiltonian-inspired “partial” mixers.

Conclusion and Summary

- QAOA(+) offers a fairly **general framework** for combinatorial optimization.
- **Interplay** between
  - Mixers,
  - Phase Separators,
  - State preparation.
- **Runtime** determined by
  - Circuit implementation,
  - Scaling in the number of rounds p,
  - Complexity of parameter finding procedure.
- **NISQ-friendly** only for hardware-native Quadratic Unconstrained Binary Optimization problems, though some error mitigation strategies exist.

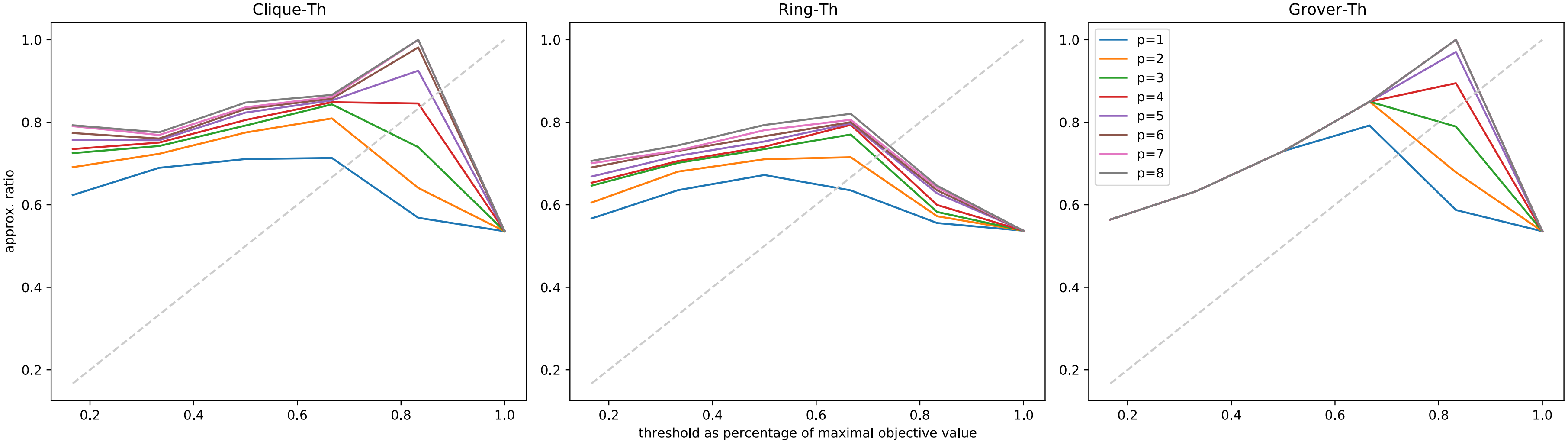
Deep Dive:
Hamming weight constraint

Phase Separator design

Phase Separators phase states $|x\rangle$ by a function of their cost: $\boldsymbol{U_P(\gamma)|x\rangle = e^{-i\,\gamma\, f(C(x))}|x\rangle}$

- **Objective value phase separator:** $f(C(x)) = \mathrm{C}(x)$
- **Threshold value phase separator:** $f(C(x)) = \delta_{th}(C(x)) = 1$ iff $\mathrm{C}(x) > th$, else 0.

For Grover Mixer, for p fixed, find threshold th with **binary search** with $\beta_i = \gamma_i = \pi$.
Same for other mixers, but angle finding there is more difficult.



Circuit implementations

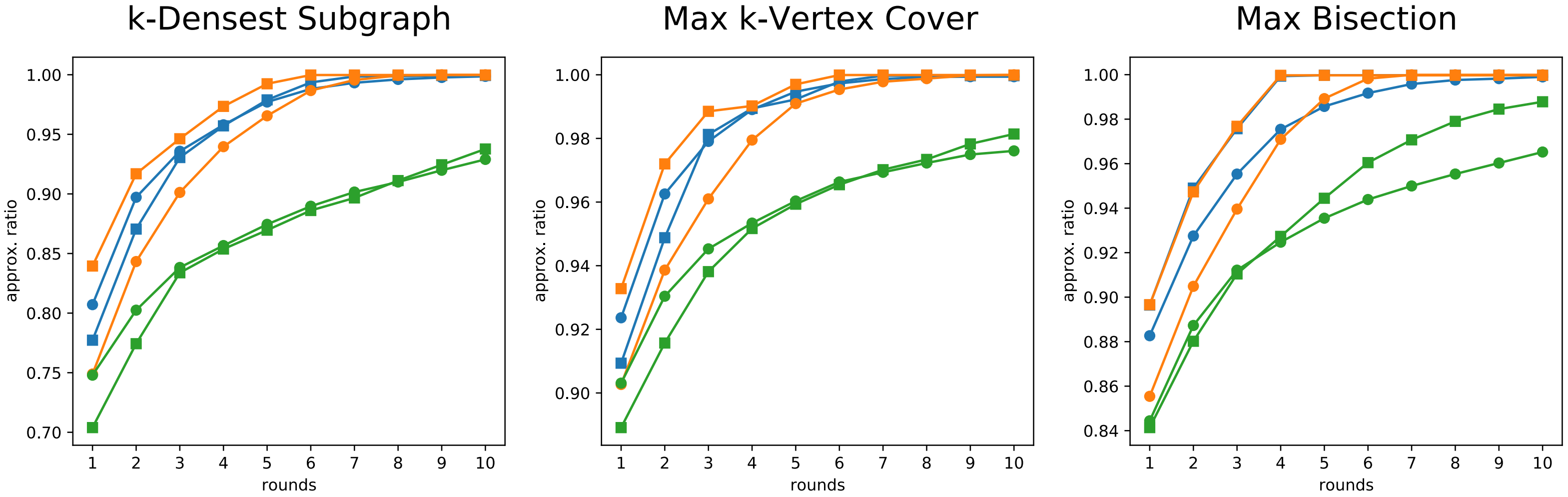
Graph Size n, Hamming weight k, maximum degree D, arithmetic precision ϵ

Unitaries U_S, U_P, U_M	Depth on Linear Nearest Neighbor	Depth on Full Connectivity	Number of Ancillas	Comment
$U_S\|0\rangle =$ Dicke State $\|D_k^n\rangle$	$\mathcal{O}(n)$	$\mathcal{O}(k \log \frac{n}{k})$	–	
Ojective-V Phase Separator	$\mathcal{O}(n)$	$\mathcal{O}(D)$	–	
Threshold Phase Separator	$\mathcal{O}(n^2 \log n)$ $\mathcal{O}(n \log n)$	$\mathcal{O}(n^2 \log n)$ $\mathcal{O}(D \log n)$	$\mathcal{O}(\log n)$ $\mathcal{O}(n \log n)$	compute objective value to compare to threshold th.
Ring Mixer $H_M = \sum_{j=i+1} X_iX_j + Y_iY_j$	– $\mathcal{O}(n)$	$\mathcal{O}(\log n)$ –	– –	FFFT $n = 2^l$, k or $n - k$ odd Givens Rotation Network
Clique Mixer $H_M = \sum_{j>i} X_iX_j + Y_iY_j$	$\mathcal{O}(n \operatorname{polylog} \frac{n}{\epsilon})$ $\mathcal{O}(n^3 \log n)$	$\mathcal{O}(n \operatorname{polylog} \frac{n}{\epsilon})$ $\mathcal{O}(n^3 \log n)$	$\mathcal{O}(\operatorname{polylog} \frac{n}{\epsilon})$ $\mathcal{O}(\log n)$	Schur Transform to get decomposition into total angular momentum and z-component.
Grover Mixer $H_M = \|D_k^n\rangle\langle D_k^n\|$	$\mathcal{O}(n)$	$\mathcal{O}(n)$ $\mathcal{O}(k \log \frac{n}{k})$	– $\mathcal{O}(\frac{n}{k})$	

Performance for fixed *n*

Approximation ratio average over 12 random $G_{n,p=0.5}$ graph instances for different Mixer / Phase Separator combinations with $k = \frac{n}{2}$.

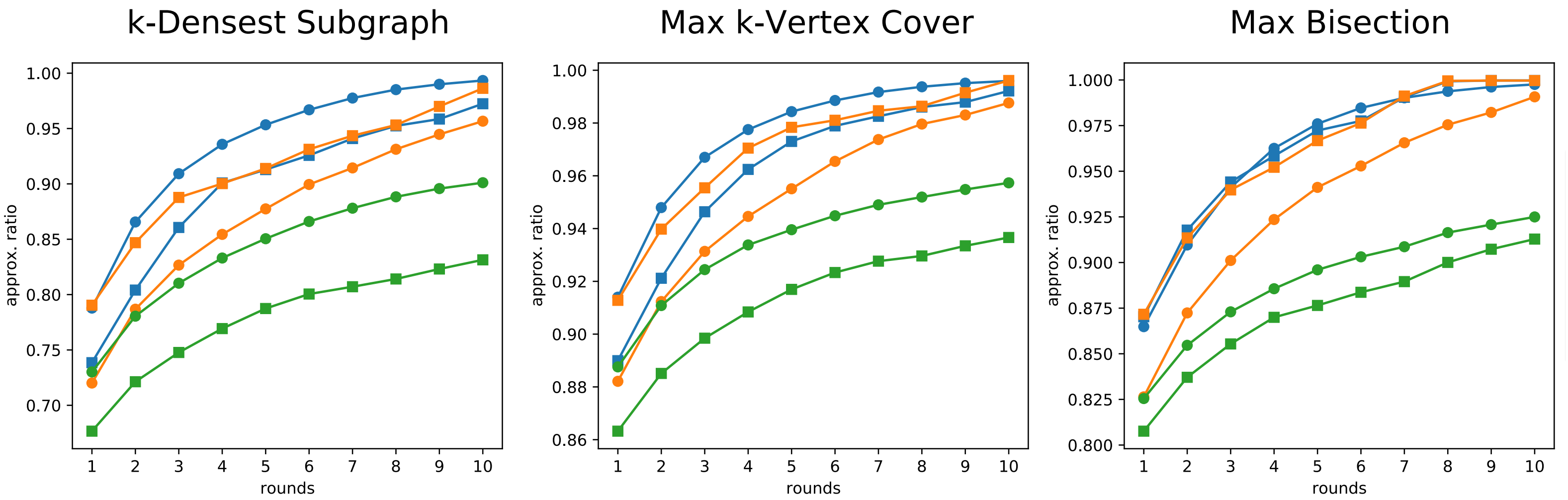
n=8



Performance for fixed *n*

Approximation ratio average over 12 random $G_{n,p=0.5}$ graph instances for different Mixer / Phase Separator combinations with $k = \frac{n}{2}$.

n=10

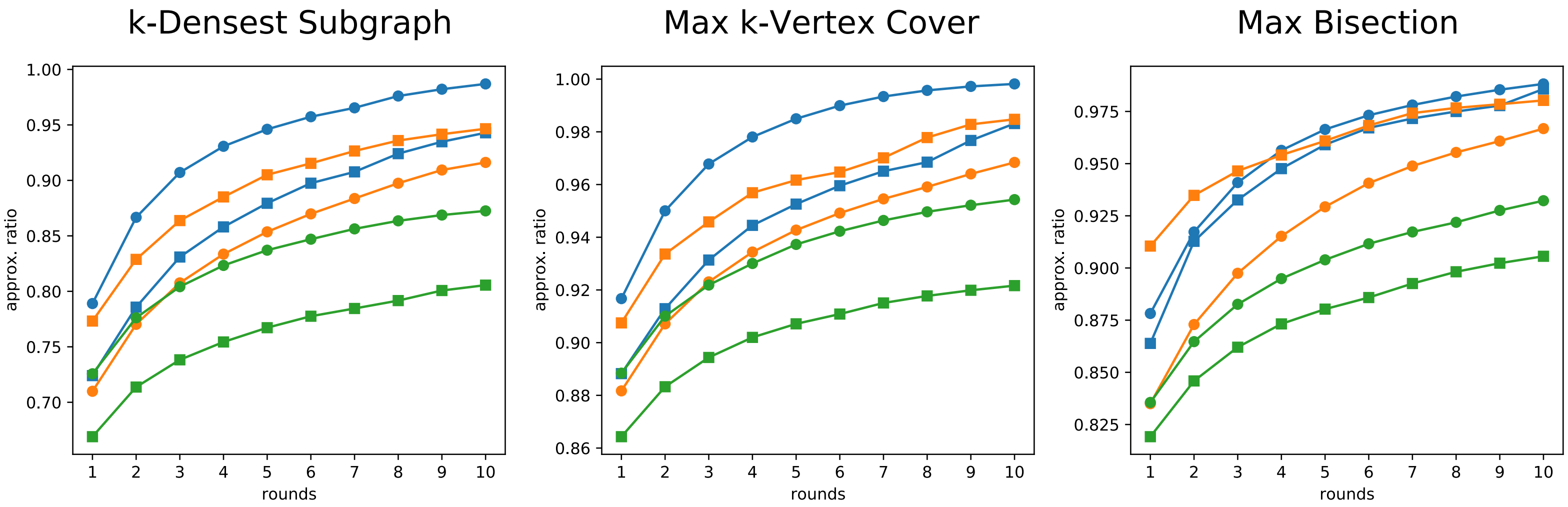


Los Alamos NATIONAL LABORATORY

Performance for fixed *n*

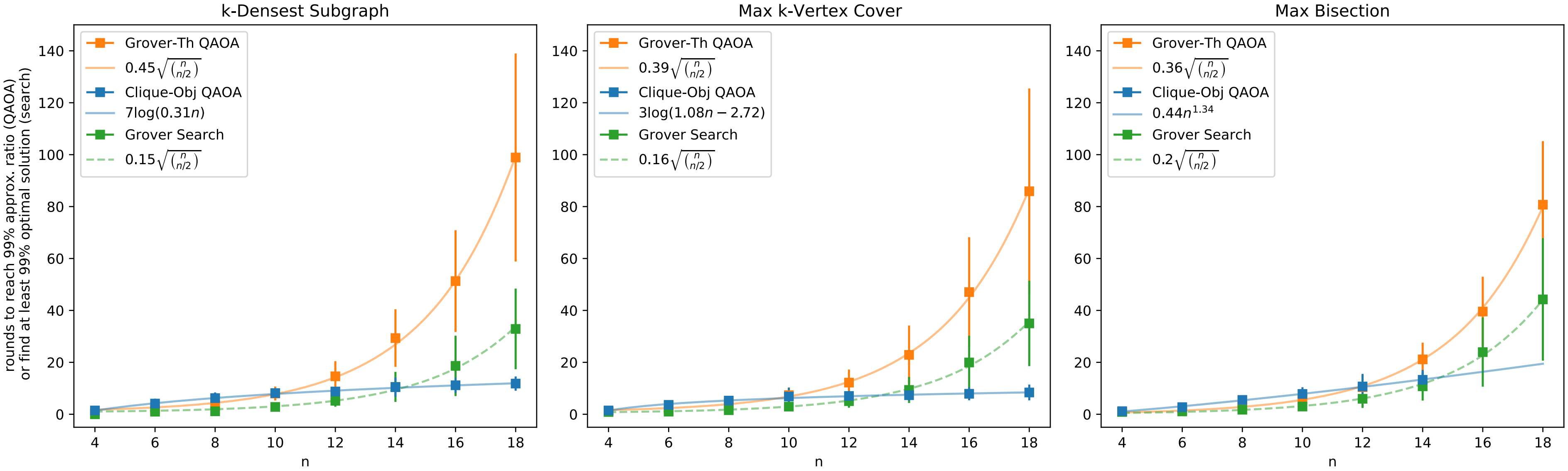
Approximation ratio average over 12 random $G_{n,p=0.5}$ graph instances for different Mixer / Phase Separator combinations with $k = \frac{n}{2}$.

n=12



Performance for growing *n*

How does the necessary number of rounds to reach 99% approximation ratio scale?



QAOA is above $\langle \beta, \gamma | H_C | \beta, \gamma \rangle - 1$ after $m \log m$ samples with probability $1 - {}^{1}/_{m}$.
How many rounds would a Grover Search with marked states above take?

Thank you!

Los Alamos
NATIONAL LABORATORY