Study network-related optimization problems using quantum alternating optimization ansatz

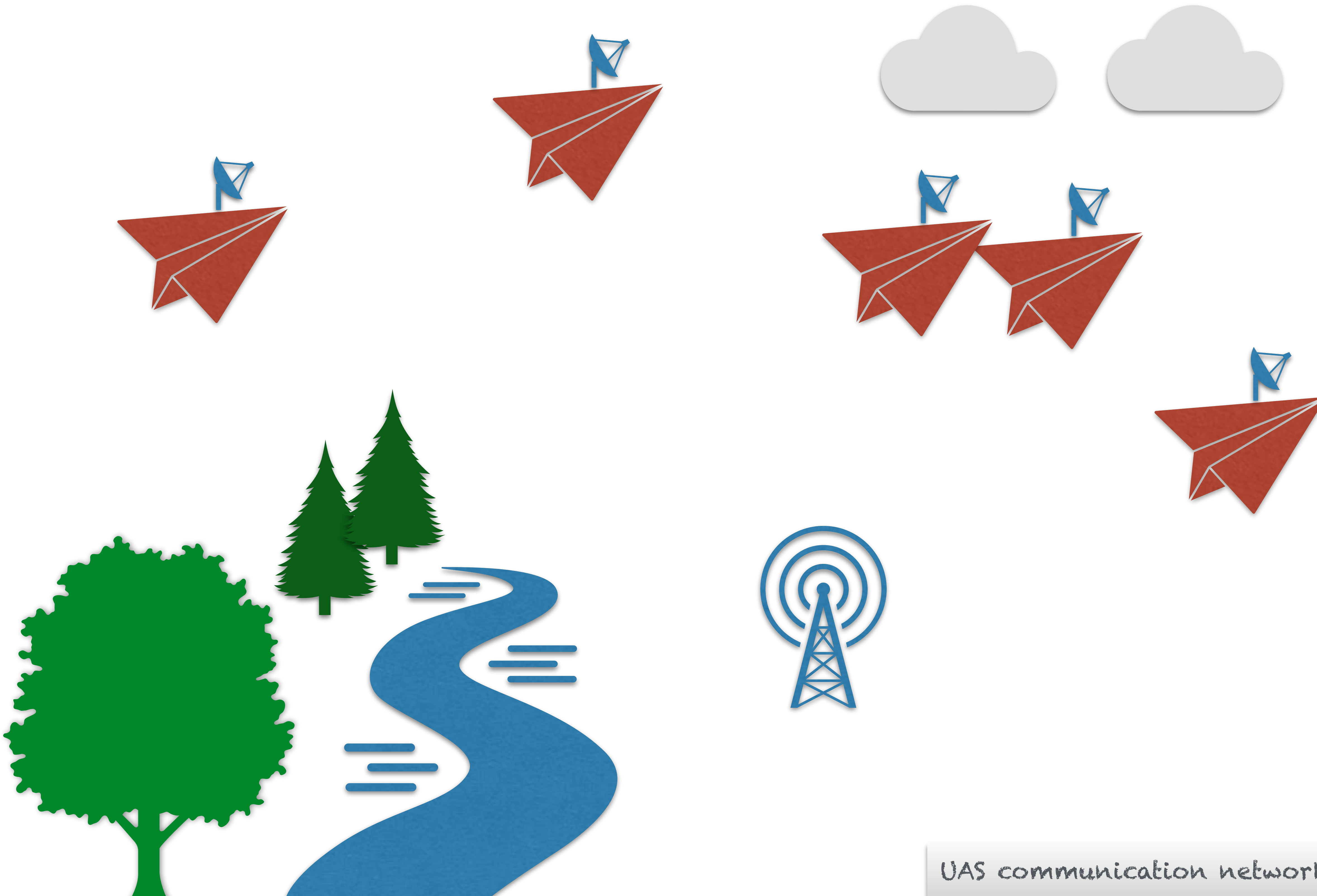
Zhihui Wang

Eleanor Rieffel, Bryan O'Gorman, Mustafa Adnane, Riccardo Mengoni, Stuart Hadfield, Davide Venturelli

Quantum AI Lab (QuAIL), USRA, NASA Ames Center

APS March Meeting 2019

zhihui.wang@nasa.gov



UAS communication network problem
NASA QuAIL

Relevance of spanning-tree/graph problem to applications

- Network connectivity problem is key to many applications.
  We study quantum heuristics (QAOA) on some network problems.

  Two basic models:
  - Spanning tree, a super basic model: minimum spanning tree. A spanning tree guarantees communication, and further cost can be minimized over the set of spanning trees.
  - Spanning graph, a model for robustness: while maintain an instantaneous spanning tree ensures connectivity at the moment, when temporal relation is considered, extra connectivity may be in favor for an overall optimal network.

  A variety of cost function based on constraints can piggyback on these two models.

Relevance of spanning-tree/graph problem to applications

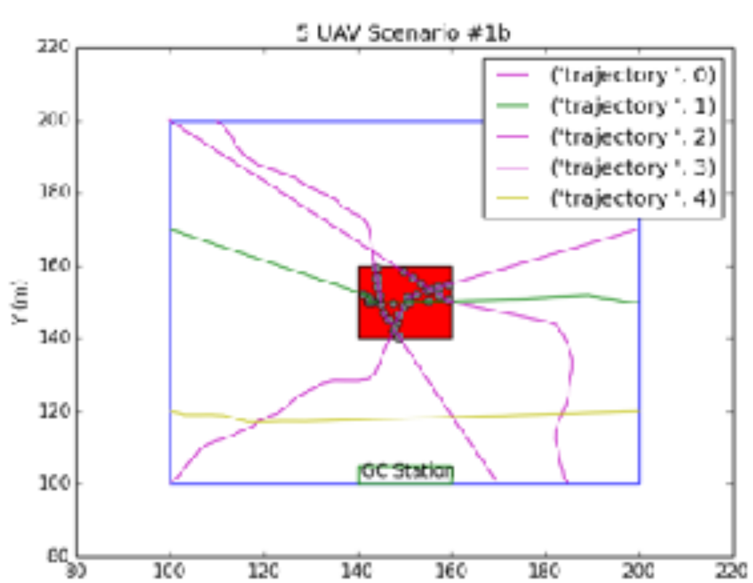
Example Testing model: Pre-determined network disruption area

- known trajectory and know disruption areas *S*
- Vehicle outside *S* can talk directly to ground control station
- Vehicle inside *S* will need to communicate with vehicles not in *S* to talk to the ground station

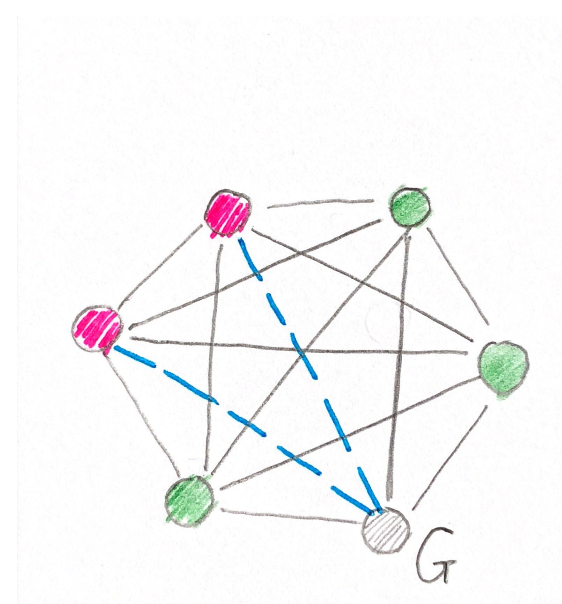
The requirement “stay connected” is equivalent to the connectivity-problem for graph *G*, in which each vehicle (as well as the ground station) corresponds to a node, and an edge exist between two vehicles can directly communicate.

Questions:

- What is the minimum communication distance? — minimizing edge weight on a **spanning tree** of *G*
- Over time, if each time the routing is changed, a cost is incurred, what is the least expensive routing plan? — **spanning subgraph** could be favorable.



Initial set of trajectories for a 5 UAV scenario for testing the quantum algorithms.



Red nodes: in disruption area
Green nodes: not in disruption area
Grey node: ground station
Blue lines: missing connections, not included in the graph

Ames Research Center

Quantum Approximate Optimization Algorithms (QAOA)

- Quantum Heuristics suitable for near-term quantum hardware
- Hard Constraints: have to be satisfied in a solution
- Soft Constraints: cost to be optimized
- Underlying idea: search in a ***feasible*** subspace
  - Start from a ***feasible*** state: one that satisfies all hard constraints
  - Mixing (create coherent superposition) of all feasible states
- Challenges:
  - Come up with classical mixing rules that maintains the feasible subspace and corresponding quantum mixing operators
  - Acquire a good set of parameters (angles)

[Farhi, Goldstone, and Gutmann, '14]

[Hadfield, arXiv 1709.03489]

One formula summary of the algorithm

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

Mixing: explore the solution space   Cost: associate with problem characteristics

Mixer for the Spanning Tree Problem

- Hard constraints:

  the solution being a spanning tree. — No loops

  Mixer:

  Basic move: For a node *v*, swap its parents between (*p*, *q*),

  conditioned on *p*, *q* are not descendants of *v.*

  Realized through manipulation of two binary matrices.

  Descendent Matrix ***D***: whether *u* is *v*'s descendant.

  Parent Matrix ***A***: whether *u* is *v*'s parent.

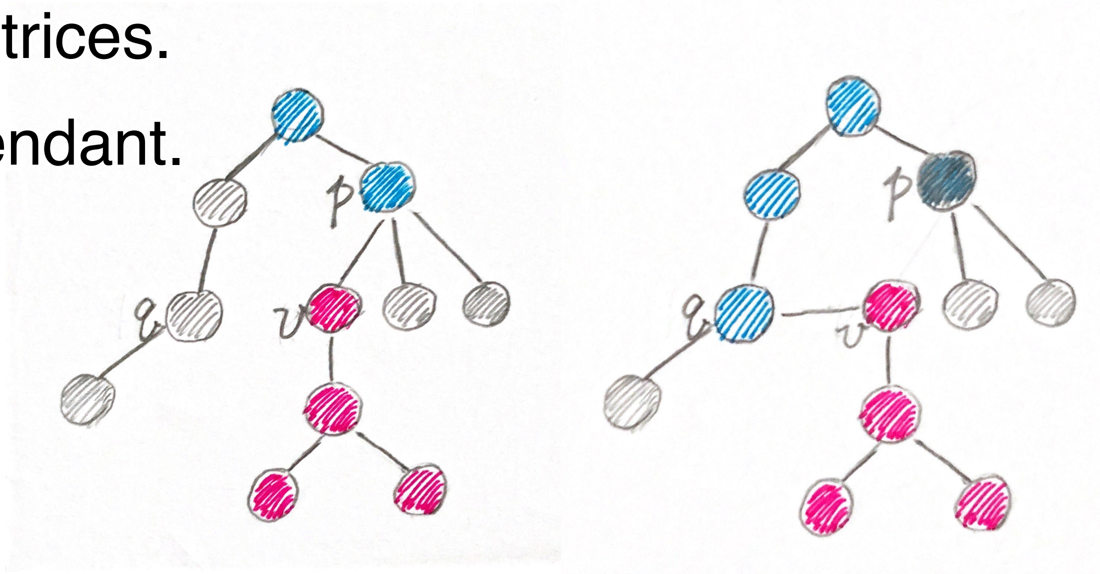
  Reversible move:

  Swap parents of *v*

$$\Lambda[\bar{D}_{vp} \wedge \bar{D}_{vq}]\ \mathrm{SWAP}(A_{pv}, A_{qv})$$

  The descendant relation between any descendants of v and any (ancestor of *p* XOR ancestor of *q*), is negated.

$$\Lambda\left[D_{v,l} \wedge (D_{kp} \neq D_{kq}) \wedge \bar{D}_{vp} \wedge \bar{D}_{vq} \wedge (A_{pv} \neq A_{qv})\right] \mathrm{NOT}(D_{kl})$$



Mixer for the Spanning Tree Problem

For each (v, p, q)

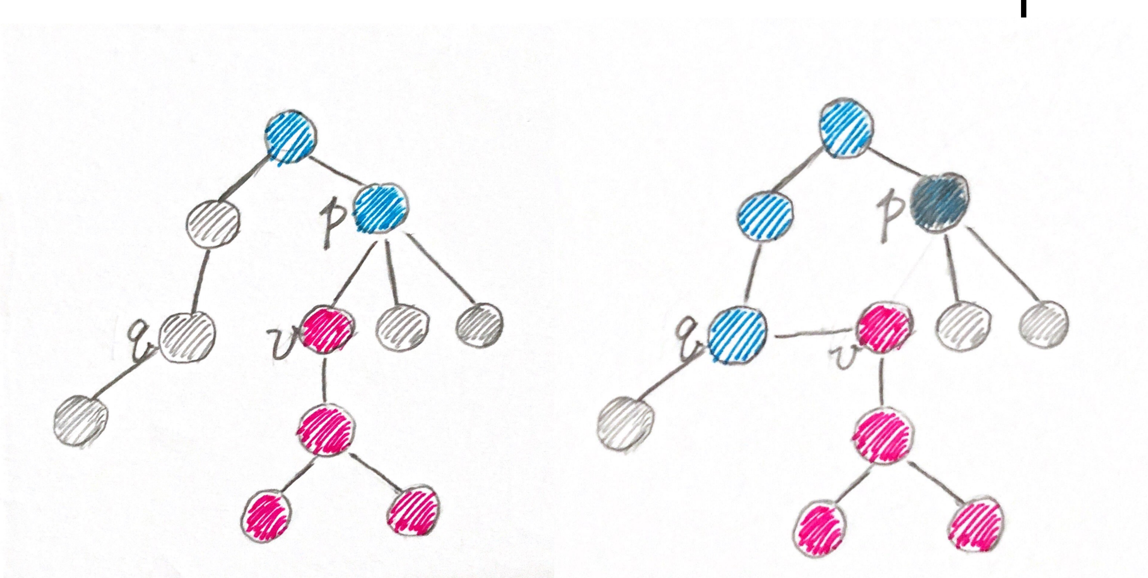
Classical operation	**Quantum operators**
$\Lambda[\bar{D}_{vp} \wedge \bar{D}_{vq}]$ SWAP(A_{pv}, A_{qv})	$M^{(A)}_{v;p,q} = \lvert 00\rangle\langle 00 \rvert_{D_{vp}D_{vq}} \otimes (\lvert 01\rangle\langle 10\rvert + \lvert 10\rangle\langle 01\rvert)_{A_{pv},A_{qv}}$

Further for each (k, l)

$\Lambda\Big[D_{v,l} \wedge (D_{kp} \neq D_{kq}) \wedge \bar{D}_{vp} \wedge \bar{D}_{vq} \wedge (A_{pv} \neq A_{qv})\Big]$ NOT(D_{kl})	$M^{(D)}_{v;p,q;k,l} = P_{A_{pv}A_{qv}} \otimes \lvert 00\rangle\langle 00\rvert_{D_{vp}D_{vq}} \otimes \Big(Q_{D_{kp}D_{kq}} \otimes \lvert 1\rangle\langle 1\rvert_{D_{vl}} \otimes X_{D_{kl}} + (I - Q_{D_{kp}D_{kq}} \otimes \lvert 1\rangle\langle 1\rvert_{D_{vl}}) \otimes I_{D_{kl}}\Big)$ where $P \equiv \lvert 10\rangle\langle 10\rvert + \lvert 01\rangle\langle 01\rvert$ and $Q \equiv I - \lvert 00\rangle\langle 00\rvert$

Mixing operator for (v, p, q): A and D updated together

$$
\begin{aligned}
H^M_{v;p,q} &= M^{(A)}_{v;p,q} \prod_{(k\neq v, l\notin\{p,q\})} M^{(D)}_{v;p,q;k,l} \\
&= |00\rangle\langle 00|_{D_{vp}D_{vq}} \\
&\quad \otimes(|01\rangle\langle 10| + |10\rangle\langle 01|)_{A_{pv},A_{qv}} \\
&\quad \otimes \prod_{(k\neq v, l\notin\{p,q\})} \Big(Q_{D_{kp}D_{kq}} \otimes |1\rangle\langle 1|_{D_{vl}} \otimes X_{D_{kl}} \\
&\quad + (I_{D_{kp}D_{kq}} I_{D_{vl}} - Q_{D_{kp}D_{kq}} \otimes |1\rangle\langle 1|_{D_{vl}}) \otimes I_{D_{kl}} \Big)
\end{aligned}
$$



Mixer for the **Spanning Graph** Problem

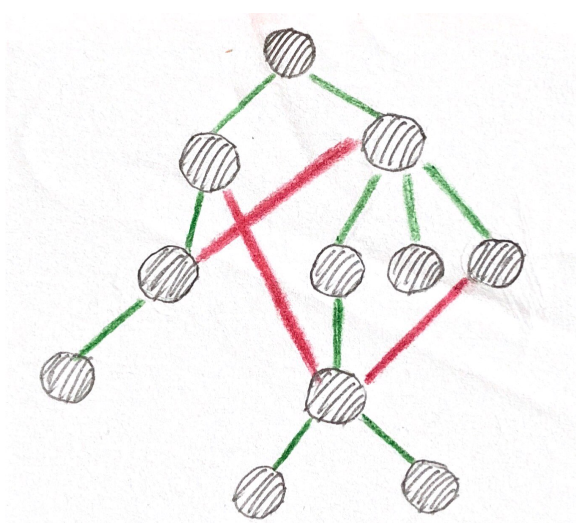
Spanning tree + more connectivity   Potential applications: more robust network

Underlying idea for a mixer:

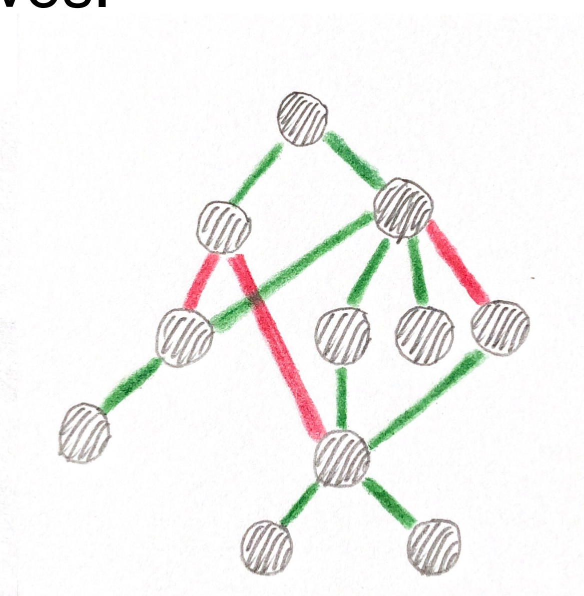
Each spanning graph *G'* can be partitioned into a spanning tree ***T***, and the set of extra edges ***Ø***. There are multiple partitions.

With a spanning tree *T* fixed, adding/removing different sets of extra edges (edges that are not in T) lead to different spanning graphs.

Therefore, mediated by the a mixer for spanning tree, we can explore all spanning graphs through these two moves.



$E(G') = (\mathbf{E(T)}, \boldsymbol{\emptyset})$



$E(G') = (\mathbf{E(T')}, \boldsymbol{\emptyset'})$

- Mixer between spanning graphs with the same backbone spanning tree

$$G' = (T, \Phi) \longleftrightarrow G'' = (T, \Phi')$$

For each edge $\{u, v\}$

$$H_{(u,v)}^{(\Phi)} = \underbrace{|00\rangle\langle 00|_{A_{uv}A_{vu}}}_{\text{edge } \{u,v\} \text{ not in the tree}} \otimes \underbrace{X_{\Phi_{u,v}}}_{\text{add/remove } \{u,v\} \text{ from } \Phi}$$

Corresponding unitary

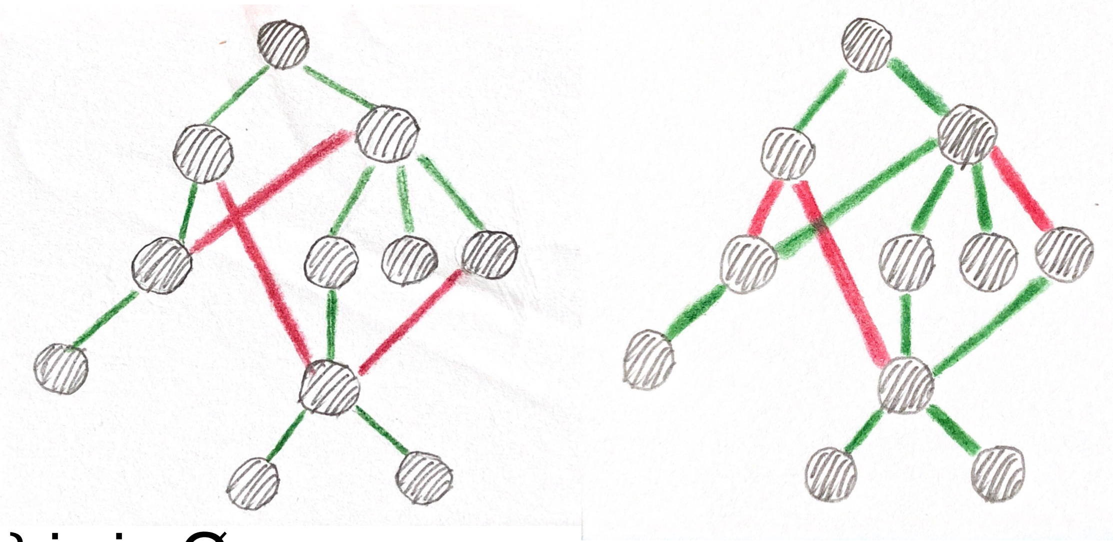
$$\begin{aligned} U_{(u,v)}^{(\Phi)}(\beta) &= \exp[-i\beta H_{(u,v)}^{(\Phi)}] \\ &= |00\rangle\langle 00|_{A_{uv}A_{vu}} \otimes e^{-i\beta X_{\Phi_{u,v}}} \\ &\quad +(I_{A_{uv},A_{v,u}} - |00\rangle\langle 00|_{A_{uv},A_{v,u}}) \otimes I_{\Phi_{u,v}} \end{aligned}$$

Total: $m = |E|$ doubly controlled single-qubit gates, each $\sim$ 12 C-NOT gates

- Mixer between partitions of a spanning $G' = (T, \Phi) = (T', \Phi')$

  Quantum register: $|D, A, \Phi\rangle$

  G' and *G* share spanning-trees, thus the spanning-tree mixer can be applied to any *G*'



Modification in parents swapping (v, p, q):

Consider *p to be parent* of *v* only when edge {*v*, *p* } is in Ø;
Disable a parental relation between q and v, and place the edge {*v*, *q* } in Ø;
i.e., *E*(*T*) and Ø swaps elements {*v*, *p* } and {*v*, *q* }

Mixing Hamiltonian:

$$H^M_{(v,p,q)} = \underbrace{|00\rangle\langle 00|_{D_{vp}D_{vq}}}_{} \otimes M^{(A\Phi)}_{pv,qv} \otimes M^{(D)}_{v,p,q}$$

neither *p* or *q* is descendant of *v*

$$M^{(A\Phi)}_{pv,qv} \equiv (|0110\rangle\langle 1001| + |1001\rangle\langle 0110|)_{A_{pv},A_{qv},\Phi_{pv},\Phi_{qv}}$$ ***E*(*T*) and Ø swaps edges (*v*, *p*) and (*v*, *q*)**

$$M^{(D)}_{v,p,q} \equiv \prod_{\underbrace{(k\neq v,\ l\notin\{p,q\})}_{}} M^{(D)}_{v;p,q;k,l}$$ **For each (*v,p,q*), all *(k,l*) need to be updated together.**

$$M^{(D)}_{v;p,q;k,l} = \Big(Q_{D_{kp}D_{kq}} \otimes |1\rangle\langle 1|_{D_{vl}} \otimes X_{D_{kl}}$$ **Update *D for* relevant *l's***

$$+ (I_{D_{kp}D_{kq}} \otimes I_{D_{vl}} - Q_{D_{kp}D_{kq}} \otimes |1\rangle\langle 1|_{D_{vl}}) \otimes I_{D_{kl}}\Big)$$ **Identity for irrelevant *l's***

Mixing Unitary:

$$
\begin{aligned}
U^{M}_{(v,p,q)}(\beta) &= \exp(-i\beta H^{M}_{(v,p,q)}) \\
&= \underline{|00\rangle\langle 00|_{D_{vp}D_{vq}}} \otimes \exp\left[-i\beta M^{(A\Phi)}_{pv,qv} \otimes M^{(D)}_{v,p,q}]\right)\Big] \\
&\quad +\big(I - |00\rangle\langle 00|_{D_{vp}D_{vq}}\big) \otimes I_{\text{others}}
\end{aligned}
$$

double-control

$$M^{(A\Phi)}_{pv,qv} \equiv (|0110\rangle\langle 1001| + |1001\rangle\langle 0110|)_{A_{pv},A_{qv},\Phi_{pv},\Phi_{qv}}$$

—> 4 qubits gates: XXYY + XYXY + XYYX +…

$$M^{(D)}_{v,p,q} \equiv \prod_{\underline{(k\neq v,\ l\notin\{p,q\})}} M^{(D)}_{v;p,q;k,l}$$

For each (*v,p,q*), all (*k,l*) need to be updated together.
Leads to high-locality of physical qubits and deep circuit depth

$$
\begin{aligned}
M^{(D)}_{v;p,q;k,l} = \Big(&Q_{D_{kp}D_{kq}} \otimes |1\rangle\langle 1|_{D_{vl}} \otimes X_{D_{kl}} \\
&+\big(I_{D_{kp}D_{kq}} \otimes I_{D_{vl}} - Q_{D_{kp}D_{kq}} \otimes |1\rangle\langle 1|_{D_{vl}}\big) \otimes I_{D_{kl}}\big)\Big)
\end{aligned}
$$

single-qubit operation controlled by a 3-q projector

$$H^{(\Phi)}_{(u,v)} = |00\rangle\langle 00|_{A_{uv}A_{vu}} \otimes X_{\Phi_{u,v}}$$

Resource requirement

Number of qubits $N_{qubit} = (n-1)^2 + 2m + m$

Qubits connectivity

Total number of connection between qubits: $\sim n^4$

Maximum qubit degree: $\sim n^2$

— High locality caused by simultaneous update of $\sim n^2$ elements of D

Break-up:

D: form a clique of size $(n-1)^2$

$D - A, \Phi$: Each D_{uv} is connected to all elements of A, and Φ

Therefore, each element of D is connected to all other qubits

$\rightarrow$ Num. links: $(n-1)^2(N_{\text{qubit}} - 1) \sim n^4$

$A_{pv}A_{qv} \rightarrow$ All edges injecting to vertex v (elements of column A_v) form a clique of size d_v. Num. links: $\sum_v \frac{d_v(d_v-1)}{2}$

$A_{uv}A_{vu} \rightarrow$ A link exists between A_{uv} and A_{vu} for any edge $\{u, v\}$. Num links: m

$A_{pv}\Phi_{qv}$ and $A_{pv}\Phi_{pv} \rightarrow$ A link exists between A_{pv} and Φ_{qv} for all q. Num. links: $\sum_v d_v^2$

$\Phi_{pv}\Phi_{qv} \rightarrow$ All edges that connects v form a clique of size d_v. Num. links: $\sum \frac{d_v(d_v-1)}{2}$

Resource requirement, example

Assume all-to-all connectivity between physical qubits on hardware, and a complete problem graph

n	Num. of physical qubits required
3	13
4	27
5	46
6	70
7	99
8	133
9	172
10	216

Application relevant (demo number of UAV's) problem size
Current / near-term gate-model quantum hardware size

Summary

QAOA for constrained optimization problem could greatly benefit from using a problem-dependent mixer.

We developed a QAOA mapping for the problem of degree-constrained spanning-tree and spanning-graph problems that could be feasible to test on NISQ devices.

Next:

simulation;

test runs on quantum hardware

performance analysis