Ames Research Center

Exploring network-related optimization problems using quantum heuristics

Zhihui Wang[1,2], Eleanor G. Rieffel[1], Mustafa Adnane[1], Bryan O'Gorman[1], Riccardo Mengoni[1], Stuart Hadfield[1,2], Davide Venturelli[1,2]
[1] Quantum Artificial Intelligence Lab (QuAIL) NASA Ames, [2]Universities Space Research Association

1. Study network problems with QA and QAOA

Network-related connectivity optimization problems are underlying a wide range of applications and are also of high computational complexity.
We consider studying network optimization problems using two types of quantum heuristics.
One is quantum annealing, and the other Quantum Alternating Operator Ansatz, an extension of the Quantum Approximate Optimization Algorithms for gate-model quantum computation, in which a cost-function based unitary and a non-commuting mixing unitary are applied alternately.

We present problem mappings for problems of finding the spanning-tree or spanning-graph of a graph that optimizes certain costs, and a variant that further requires the spanning-tree be degree-bounded.

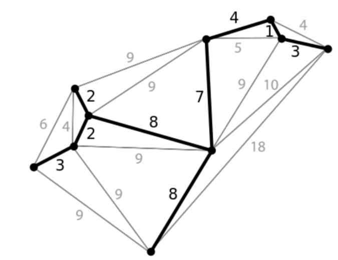
2. The spanning-tree problem

Given graph $G = (V, E)$ with weights associated with each edge

Find $E' \in E$ such that $G' = (V, E')$ form a tree

and the weighted sum of edges is minimized

NP hard: if the maximum vertex degree in the tree is bounded.



3. Mapping to QUBO using penalty terms

- All constraints are cast into penalty terms in the cost Hamiltonian

- Level-based mappings

Randomly fix a node to be the root of the tree

Level: distance to the root in the tree.

Variables: $x_{p,v} = \begin{cases} 1 & p \text{ is parent of } v \\ 0 & p \text{ is not parent of } v \end{cases}$ $y_{v,l} = \begin{cases} 1 & v \text{ is on level } l \\ 0 & v \text{ is not on level } l \end{cases}$

Each node *v* (other than root) has exactly one parent $\left(\sum_p x_{p,v} - 1\right)^2$

Each node *v* has exactly one level (only root is on level 1) $\left(\sum_{l=2}^{n} y_{v,l} - 1\right)^2$

Node v's parent must be one-level lower than *v*. $x_{p,v} y_{v,l} (1 - y_{p,l-1})$

Break cubic terms into QUBO by introducing ancilla qubits

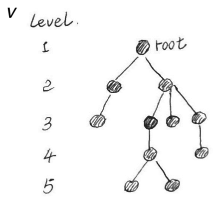
$xy \to a \qquad xyz \to az + f(a, x, y)$

$f(a, x, y) = 3a + xy - 2ax - 2ay$ ensures $a = xy$

- Ordering-based mapping

Order vertices of *G, parental relation must follow the same ordering.*

- One-hot encoding of the ordering:
  $x_{v,i} = \begin{cases} 1 \text{ iff } v \text{ is in position } i \\ 0 \text{ otherwise} \end{cases}$ $\left(\sum_i x_{v,i} - 1\right)^2 \quad \left(\sum_v x_{v,i} - 1\right)^2$
- Each vertex has exactly one parent
  $y_{p,i} = \begin{cases} 1 \text{ iff } p \text{ is parent of vertex in position } i \\ 0 \text{ otherwise} \end{cases}$ $\left(\sum_{p \in V} y_{p,i} - 1\right)^2$
- One's child must be a neighbor $\sum_{\{p,v\} \notin E(G)} \sum_i x_{v,i} y_{p,i}$
- Degree-constraint: for each *v* $\left(\sum_i y_{v,i} - \lambda_v\right)^2, \quad \lambda_v \in [0, \Delta - 1]$
  - unary encoding



- Tradeoff between QUBO mappings:
  - Tight level-based: less qubits, connections, less resource needed to embed on hardware
  - Lose level-based: bigger optimal solution space (degenerate), may facilitate easier search in QA
  - Ordering-based: no ancilla qubits required for 3rd to 2nd order reduction

4. Mapping to QAOA using a problem-specific mixer

- Quantum Approximate Optimization Algorithms (QAOA)

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

Cost: problem characteristics
Mixing: explore the solution space

- Hard Constraints: have to be satisfied in a solution
- Soft Constraints: cost to be optimized
- Underlying idea: search in a ***feasible*** subspace that satisfies all hard constraint; 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

Hard constraints:

the solution being a spanning tree. — No loops

Mixer:

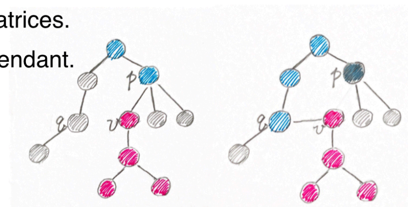
Basic move: For a node *v*, swap its parents between (*p*, *q*),

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*

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

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

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