Distributed Quantum Computing for Network Problems, and other selected topics

Eleanor Rieffel

Senior Research Scientist,

Lead, Quantum Artificial Intelligence Lab (QuAIL)

NASA Ames Research Center, Moffett Field, CA

Joint work with Phillip Kerger, David Bernal Neira, and Zoe Gonzales Izquierdo

NASA QuAIL team: Sohaib Alam, David Bernal Neira, Lucas Brady, Stephen Cotton, Shon Grabbe, Zoe Gonzalez Izquierdo, Stuart Hadfield, Aaron Lott, Salvatore Mandrà, Jeffrey Marshall, Gianni Mossi, Jason Saied, Norm Tubman, Davide Venturelli, Zhihui Wang, Filip Wudarski

Why Quantum Computing at NASA?

NASA constantly confronting massively challenging computational problems

- Computational capacity limits mission scope and aims

NASA's leading supercomputing efforts

Low energy and water use supercomputing

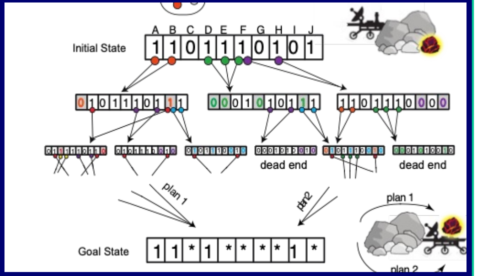
NASA QuAIL mandate:

Determine the potential for quantum computation to enable more ambitious and safer NASA missions in the future

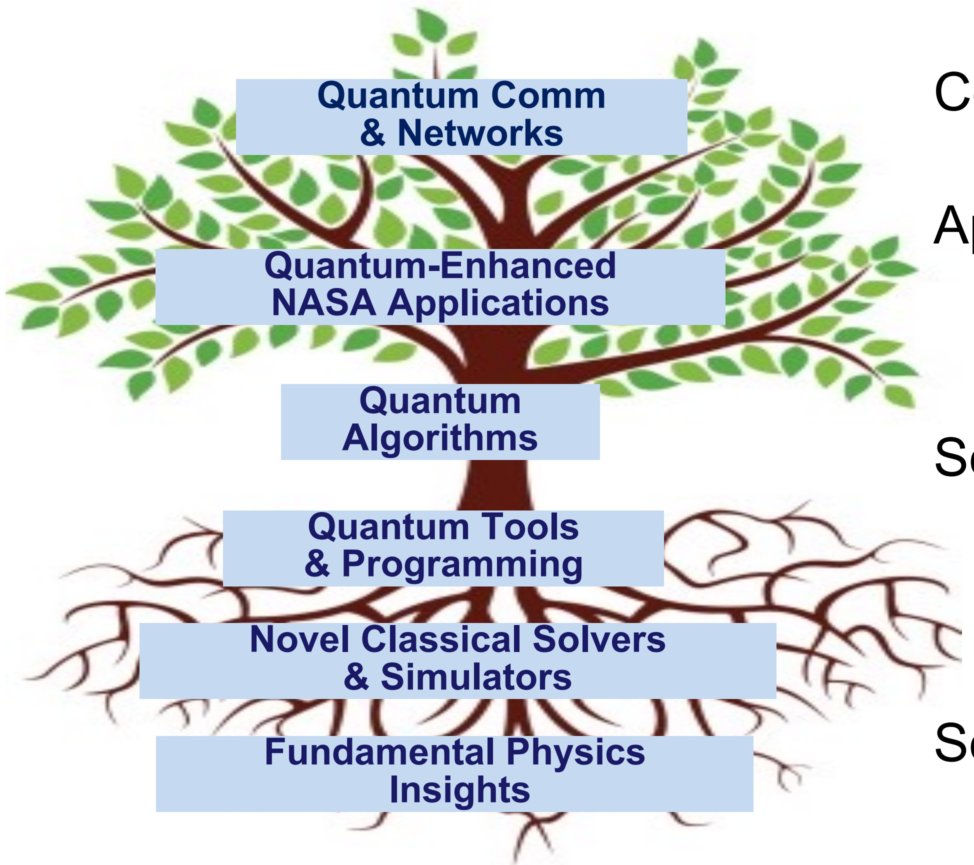
Aeronautics:
Air Traffic Management and Robust Airspace Communication

Earth Science:
Data Analysis and Image Processing

Space Exploration:
Resource Allocation and Scheduling



Quantum Computing at NASA



Communication & Networks

- Quantum networking
- Distributed QC

Application Focus Areas

- Planning and scheduling
- Material science
- Fault diagnosis
- Machine Learning
- High-energy physics

Software Tools & Algorithms

- Quantum algorithm design
- Mapping, parameter setting, error mitigation
- Hybrid quantum-classical approaches
- Compiling quantum algorithms to hardware

Solvers & Simulators

- Physics-inspired classical solvers
- HPC quantum circuit simulators

Physics Insights

- Co-design quantum hardware

E.G. Rieffel, et al. (2019) ***From Ansätze to Z-gates: a NASA View of Quantum Computing***, *Advances in Parallel Computing* **34**, 133-160

R. Biswas et al. (2017) ***A NASA Perspective on Quantum Computing: Opportunities and Challenges***, Parallel Computing, Volume 64, p. 81-98

Quantum computing has entered the NISQ Era

Quantum supremacy achieved!

- Perform computations not possible on even the largest supercomputers

Article

Quantum supremacy using a programmable superconducting processor

https://doi.org/10.1038/s41586-019-1666-5
Received: 22 July 2019
Accepted: 20 September 2019
Published online: 23 October 2019

Nature | Vol 574 | 24 OCTOBER 2019 | 505

Cover article, Nature, 24 Oct 2019

Google, NASA, ORNL collaboration

... what about useful quantum supremacy?

- Quantum processors currently too small and non-robust to be useful for solving practical problems

Uses of these still limited, quantum devices?

(1) Unprecedented opportunity to explore and **evaluate algorithms**, both quantum and hybrid quantum-classical heuristic algorithms

(2) **Investigate quantum mechanisms** that may be harnessed for computational purposes

Insights gained feed into next generation

- **quantum algorithms**
- **quantum hardware**

Early target: Optimization; Sampling & Machine Learning; Simulation of quantum systems

If someone handed you a large, fault tolerant quantum computer, what would you run?

Current status of quantum algorithms

Quantum computing can do everything a classical computer can do

and

Provable quantum advantage known for a few dozen quantum algorithms

Unknown quantum advantage for everything else

Status of classical algorithms

- Provable bounds hard to obtain
  - Analysis is just too difficult
- Best classical algorithm not known for most problems
- Empirical evaluation required
- Ongoing development of classical heuristic approaches
  - Analyzed empirically: ran and see what happens
  - E.g. SAT, planning, machine learning, etc. competitions

A handful of proven limitations on quantum computing

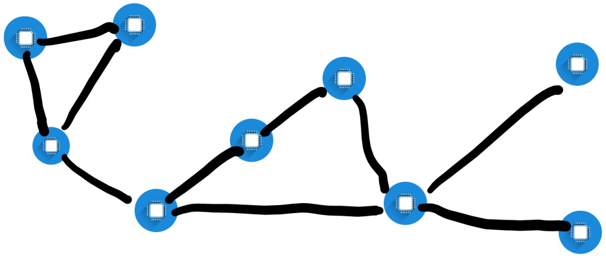
Conjecture: Quantum Heuristics will significantly broaden applications of quantum computing

Quantum Speedups for Distributed Graph Algorithms in the CONGEST-CLIQUE Model

- Introduction
  - Distributed Computation Models
  - Classical vs Quantum Versions
- Background: Prior Results
  - Negative Results in CONGEST
  - Positive Results and Contributions in CONGEST-CLIQUE
  - From Shortest Paths to Triangle Finding
  - Distributed Grover Search
- Our algorithmic Results

Distributed Graph Algorithms: Setup

- Distributed Computation: Network of Processors that communicate
  - Model as a graph where the processors are the nodes
  - Each node its own starting information, such as list of neighbors



- E.g.: Network of spacecraft, satellites, or control stations that each have computing power and can communicate
- Distributed Graph Problems: Nodes jointly answer a question by computing and communicating

CONGEST Model of Distributed Computing

Graph G = (V,E,W) with n = |V| number of nodes, m = |E| number of edges, and W the weights on the edges

CONGEST Model: Aim is to minimize the number of rounds

1. Computation happens in rounds (compute, communicate, compute, communicate, …)
2. **Congested:** Communication limited by message size: each node can send to each of its neighbors O(log(n)) bits each round
   — log(n) is length of a node id
3. Unlimited local computation at each node
4. Nodes can communicate only with their neighbors

CONGEST-CLIQUE Model:

1., 2., 3. above are the same

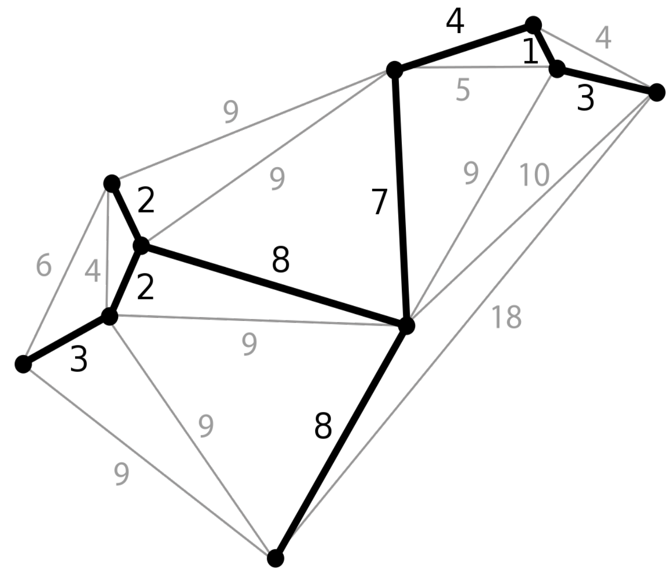
4. All nodes can communicate with each other

Quantum CONGEST and CONGEST-CLIQUE Models

- Initially: Each node knows
  - its own ID
  - the ID's of its neighbors
    - assuming ID's are 1 to n → log(n) bits to encode

- Answer a question about that graph in as few rounds as possible
  - Ex: Spanning trees, subgraph detection, shortest paths...

- **Quantum versions**: Take CONGEST or CONGEST-CLIQUE, but allow messages to consist of O(log(n)) **qubits**



A minimum spanning tree (black) for the given graph (grey)
source: https://en.wikipedia.org/wiki/Minimum_spanning_tree

CONGEST: Negative Results

Main reference: Elkin et al 2012,
"Can Quantum Communication Speed Up Distributed Computation?"

- Proved limitations for quantum CONGEST model
- No speedups possible for many fundamental problems:
  Shortest paths, Minimum Spanning Tree, Steiner Tree, Min Cut, Hamiltonian Cycle…

- Intuition: In CONGEST, a significant bottleneck can be communicating between "distant" parts of the network – qubits don't help with that

What about quantum CONGEST CLIQUE?

- Elkin et al's results and analysis **do not carry over to the CONGEST CLIQUE**
- **Can quantum communication help in this model?**

Positive results in Quantum CONGEST CLIQUE (qCC):

- **Faster Triangle Detection, Izumi & Le Gall 2019**
- **Faster All-Pairs Shortest-Paths (APSP), Izumi & Le Gall 2020**
  - This was in Elkin's list of problems not admitting speedups in the CONGEST model
- Makes use of a distributed Grover search subroutine

What other problems can see speedups in qCC?

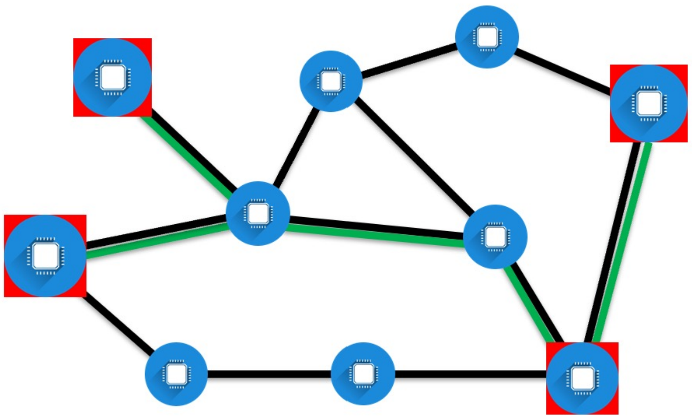
Our recent results:

- Algorithms in Quantum CONGEST-CLIQUE for
  - (approximately optimal) Steiner Trees and
  - Directed Minimum Spanning Trees

- Computation that succeeds with high probability using fewer rounds than any known classical algorithm

  $\rightarrow \tilde{O}(n^{1/4})$ versus $\tilde{O}(n^{1/3})$

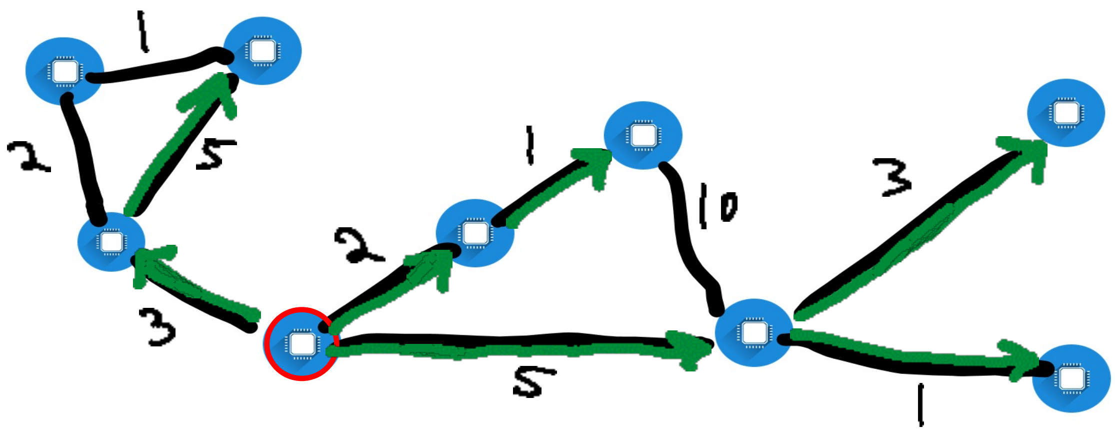
- Steiner Tree Problem (informal):
  Given a set of *terminal nodes*, find a minimum weight tree in the graph that contains them



- *Red are terminals, green is Steiner Tree*

Directed Minimum Spanning Tree:

- Given a root node r in a directed, weighted graph G, find a directed spanning tree for G rooted at r of minimum weight



A Directed Minimum Spanning Tree (green) rooted at the red node

- Techniques: Made use of fast all-pairs shortest-path via triangle finding with distributed Grover Search subroutines to obtain the faster algorithms

Algorithmic Recipe:

Make use of previous techniques such as

1. Distributed Grover Search

2. Triangle Finding

3. Distance Products

4. Shortest Paths and Routing Tables

For Steiner Trees and Directed Minimum Spanning Trees!

Distributed Grover Search

Distributed Implementation of Grover's Algorithm:

- Suppose a set X of inputs, and a node u, which we call the leader, can evaluate a function $g: X \rightarrow \{0,1\}$ in r rounds. Then there exists a quantum distributed algorithm such that u outputs an element $x \in X$ with $g(x) = 1$ using $O\left(r\sqrt{|X|}\right)$ rounds.
- Uses the same ideas as centralized Grover search: search done by leader querying other nodes
- Classical algorithm would use $O(r|X|)$ rounds in absence of additional structure
- Intuition: A node wants to inquire something related to some subset of other nodes – instead of inquiring about one node at a time, use superposition

Triangle Finding

Make use of Distributed Grover Search for finding Triangles!

Idea:

- Triangles live in $V \times V \times V$, so partition that space into n subsets
- Each node searches for triangles in its assigned partition
- Do this via sets of subsets $V_1 \times V_2 \times V_3$, where $|V_1| = |V_2| = n^{1/4}, |V_3| = \mathrm{n}^{1/2}$,

  and each subset $w \in V_1$ or V_2 contains $n^{3/4}$ nodes, and each $w \in V_3$ contains $n^{1/2}$ nodes.

- So $V_1 \times V_2 \times V_3$ has n elements, that are each a partition of nodes in $V \times V \times V$, each node gets one partition
- Making use of fast routing, the bottleneck becomes searching through the elements of V_3; Grover search can do it in $n^{1/4}$ instead of $n^{1/2}$

Distance Products

Definition 3.2. The *distance product* between two $n \times n$ matrices A and B, also sometimes referred to as the min-plus or tropical product, is defined as:

$$(A \star B)_{ij} = \min_k \{A_{ik} + B_{kj}\}. \tag{3.1}$$

- If W is the adjacency matrix of a graph, then the ij entry of $W \star W$ gives the shortest 2-hop path from node i to node j
- W^n contains shortest n-hop distances , so contains all shortest path distances (exponent w.r.t. $\star$)

Distance Products via Triangles

Definition 3.1. Given three vertices $u, v, w \in V$, we say that the triple $\{u, v, w\}$ is a negative triangle in G if $\{u, v\}$, $\{u, w\}$ and $\{v, w\}$ are edges and the inequality $f(u, v) + f(u, w) + f(v, w) < 0$ holds.

FindTriangleEdges— determines which edges are involved in at least one negative triangle

PROPOSITION 3.5. *[39] Assume that there exists a $T(n)$-round algorithm for FINDEDGES. Then there exists a $O(T(n) \log M)$-round algorithm that computes the distance product of any two $n \times n$ matrices with entries in $\{-M, \ldots, M\} \cup \{-\infty, \infty\}$.*

PROPOSITION 3.6. *Assume that there exists a $T(n, M)$-round algorithm that computes the distance product of any two $n \times n$ matrices with entries in $\{-M, \ldots, M\} \cup \{-\infty, \infty\}$. Then there exists a $O(T(n, nW) \log n)$-round algorithm for the APSP with integer weights in $\{-W, \ldots, W\}$.*

Distributed Quantum Steiner Tree Approximation Algorithm in qCC Model

Step 1 - APSP and Routing Tables: Solve the APSP problem and add an efficient routing table scheme via triangle finding in $\tilde{O}(n^{1/4})$ rounds

Step 2 - Shortest-path Forest: Construct a shortest-path forest (SPF), where each tree consists of exactly one source terminal and the shortest paths to the vertices whose closest terminal is that source terminal. This step can be completed in one round and n messages

Step 3 - Weight Modifications: Modify the edge weights depending on whether they belong to a tree (set to 0), connect nodes in the same tree (set to ∞), or connect nodes from different trees (set to distance of shortest path between root terminals of the trees that use the edge).

Step 4 - Minimum Spanning Tree: Construct a minimum spanning tree (MST) on the modified graph in $O(1)$ rounds

Step 5 - Pruning: Prune leaves of the MST that are non-terminal nodes, since these are not needed for the Steiner Tree

Distributed Quantum Minimum Spanning Tree Algorithm in qCC Model

QuantumDMST

0: To initialize, each non-root node v records its minimum incoming edge weight $W(v)$, and modifies the weights of all its other incoming edges by subtracting $W(v)$ from them. Then initialize a collection H of edges by having each non-root node add a zero-weight edge to H.

1: Have all nodes learn all edges of H, as well as the current super-vertices and active components in the graph (O(1) rounds).

2: For each connected component $H_i \subset H$, denote by $C(H_i)$ the cycle of H_i (there can only be one in each component). Let $c(H_i)$ be the node with maximal ID in $C(H_i)$, which each node can locally compute.

3: Run the algorithm for APSP and routing tables described in the previous section on this graph, or report a failure if it fails.

4: For each i, determine an edge $v_i u_i$, $v_i \notin H_i, u_i \in H_i$ minimizing $\beta_i := W_{v_i u_i} + dist(u_i, c(H_i))$, and broadcast both to all nodes in H_i.

5: Each node in each H_i applies the following updates *locally*:

- Soft-contract H_i at level β_i to soft-contract all super-vertices with distance β_i to $C(H_i)$ into one super-vertex
- Update its super-vertex ID to $c(H_i)$
- add e_i to H, effectively merging H_i with another active component of H
- for the node $c(H_i)$, update $W(c(H_i)) \leftarrow W(c(H_i)) + \beta_i$

6: Return to step 1 if H is not a single connected component. Otherwise expand H into a full DMST for G.

Main References

Elkin, M., Klauck, H., Nanongkai, D., & Pandurangan, G. (2012). *Can quantum communication speed up distributed computation?* arXiv. Retrieved from https://arxiv.org/abs/1207.5211 doi doi:10.48550/ARXIV.1207.5211

Izumi, T., & Gall, F. L. (2019, jul). Quantum distributed algorithm for the all-pairs shortest path problem in the CONGEST-CLIQUE model. In *Proceedings of the 2019 ACM symposium on principles of distributed computing.* ACM. Retrieved from https://arxiv.org/abs/1906.02456 doi: doi:10.1145/3293611.3331628

Izumi, T., Le Gall, F., & Magniez, F. (2020). *Quantum distributed algorithm for triangle finding in the congest model.* Schloss Dagstuhl - Leibniz-Zentrum für Informatik. Retrieved from https://drops.dagstuhl.de/opus/volltexte/2020/11884/ doi: doi:10.4230/LIPICS.STACS.2020.23

Saikia, P., & Karmakar, S. (2019). *Distributed approximation algorithms for steiner tree in the congested clique.* arXiv. Retrieved from https://arxiv.org/abs/1907.12011 doi: doi:10.48550/ARXIV.1907.12011

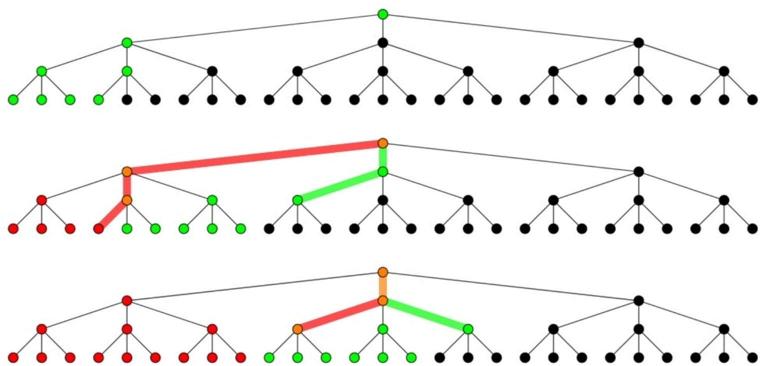
Image Credits (processors, rockets, satellites) to Pixabay (copyright-free images!), https://pixabay.com/

Please see our poster and talk with Phillip Kerger and David Bernal Neira as well!

Brief Glimpse of Other QC & OR work: CP

Quantum-accelerated Constraint Programming

- In constraint programming (CP), problems are solved with backtracking tree search augmented by logical inference
- Quantum algorithms can **accelerate the inference process** being performed at each node in the tree
- These quantum inference algorithms can then be integrated within classical, fully-quantum, or partially-quantum backtracking tree search schemes
- Partially quantum backtracking schemes yield speedups for smaller sections of the tree, intended for early, more resource-constrained quantum devices



Booth, Kyle EC, Bryan O'Gorman, Jeffrey Marshall, Stuart Hadfield, and Eleanor Rieffel. **Quantum-accelerated global constraint filtering.** In *International Conference on Principles and Practice of Constraint Programming*, pp. 72-89. 2020
Booth, Kyle EC, Bryan O'Gorman, Jeffrey Marshall, Stuart Hadfield, and Eleanor Rieffel. **Quantum-accelerated constraint programming**. *Quantum* 5 (2021): 550.

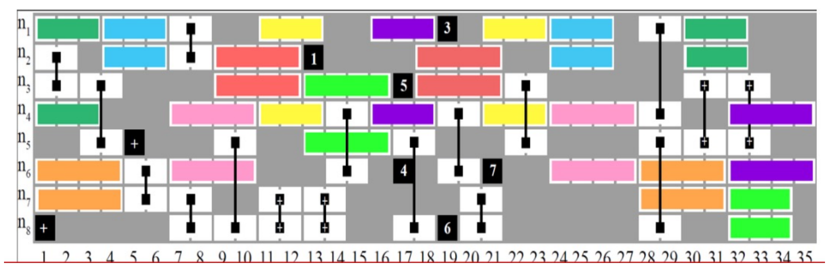
Brief Glimpse of Other QC & OR work: Qubit Routing

Quantum algorithms must be compiled before they can be run on quantum hardware

- Gate synthesis: rewrite only in terms of native gates
- Qubit routing: move information around to where two-qubit gates can act, given connectivity constraints

Can be viewed as a temporal planning problem

- Applied state of the art temporal planners
- There is now a quantum circuit compilation domain in the International Planning Competitions temporal planning track
- Combining CP with temporal planning is advantageous



Advantages of this approach

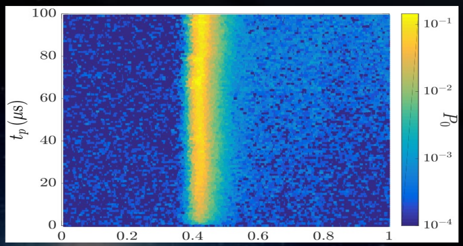
- Takes advantage of gates commuting with each other, e.g. QAOA
- Minimizes “makespan,” the time it takes to carry out the computation

D Venturelli, M Do, E. Rieffel, J Frank, Compiling quantum circuits to realistic hardware architectures using temporal planners. Quantum Science and Technology 3 (2), 2018

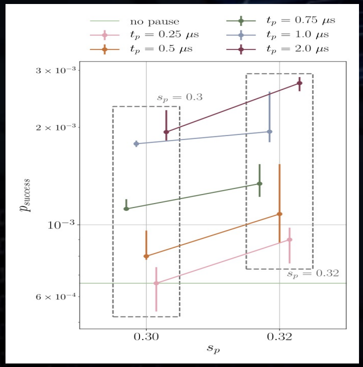
Venturelli, Davide, Minh Do, Bryan O'Gorman, Jeremy Frank, Eleanor Rieffel, Kyle EC Booth, Thanh Nguyen, Parvathi Narayan, and Sasha Nanda. "Quantum circuit compilation: An emerging application for automated reasoning." (2019).

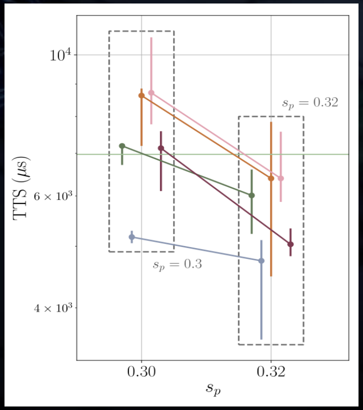
Booth, Kyle EC, Minh Do, J. Christopher Beck, Eleanor Rieffel, Davide Venturelli, and Jeremy Frank. "Comparing and integrating constraint programming and temporal planning for quantum circuit compilation." In Twenty-Eighth international conference on automated planning and scheduling. 2018

The Power of Pausing in Q. Annealing

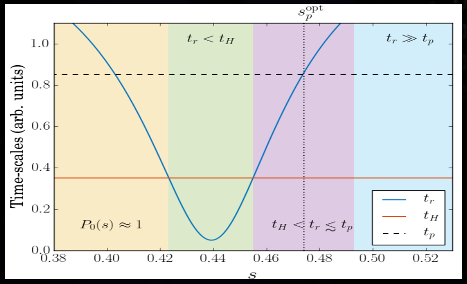


Heat map of probability of solution, depending on pause location and pause length





Anneal time regions. *Purple region is where pause is expected to be effective.*



An appropriately placed pause can improve TTS as well as the probability of success

- on embedded instances of application interest
- consistent region in which pausing helps

J. Marshall *et al.* (2019), Power of pausing: Advancing understanding of thermalization in experimental quantum annealers, *Phys. Rev. Applied* **11**, 044083

Z. Izquierdo *et al.* (2021), Ferromagnetically shifting the power of pausing, *Phys. Rev. Applied* **15**, 044013

HybridQ: A Hybrid Classical Simulator for Quantum Circuit Simulations

Hardware agnostic quantum simulator, designed to simulate large scale quantum circuits.

Can run tensor contraction simulations, direct evolution simulation and Clifford+T simulations using the same syntax

Features:

- Fully compatible with Python (3.8+)
- Low-level optimization achieved by using C++ and Just-In-Time (JIT) compilation with JAX and Numba,
- It can run seamlessly on CPU/GPU and TPU, either on single or multiple nodes (MPI) for large scale simulations, using the exact same syntax
- User-friendly interface with an advanced language to describe circuits and gates, including tools to manipulate/simplify circuits.

Open source code available at https://github.com/nasa/HybridQ

S. Mandrà, J. Marshall, E. G. Rieffel, R. Biswas, HybridQ: A Hybrid Simulator for Quantum Circuits, arXiv:2111.06868

Recent Improvements:

- Commutations rules are used to simplify circuits (useful for QAOA)
- Expansion of density matrices as superpositions of Pauli strings accepts arbitrary non-Clifford gates,
- Open-source (soon!) project with continuous-integration, multiple tests and easy installation

Brief call out to two other posters

Copositive Optimization for mixed-binary quadratic optimization via Ising solvers
Robin Brown

Proposed a hybrid classical-Ising method mixed-binary quadratic programming via convex copositive optimization and Ising solvers

- All the classical parts scale polynomially while the Ising part can be solved approximately
- Proved strong duality coming from Burer's reformulation under mild assumptions
- Our approach is beating state-of-the-art without fine tuning

Benchmarking the Operation of Quantum Heuristics and Ising Machines
David Bernal Neira

- Guidelines for performance evaluation of solvers, particularly unconventional ones such as Quantum Annealers and Coherent Ising Machines.
- Benchmarking accounting for all resources related to real-world deployment
  - Pre-processing, e.g. embedding
  - Parameter setting
  - Post-processing
- illustrative baseline family of parameter setting strategies that feature an exploration-exploitation tradeoff