Quantum Algorithms for Jet Clustering

Annie Wei
Joint work with Preksha Naik, Aram Harrow, Jesse Thaler
(arXiv:1908.08949)

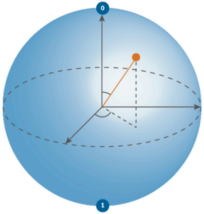
DOE CSGF Program Review

June 25 and July 1st, 2021

Outline

Quantum Computing

- Definition: Using quantum systems to perform computations.
- Challenge: How do we identify problems where we get a speedup over classical computers?
- Main approaches:
  - Analog
    - Quantum simulation
    - Quantum annealing
  - Digital: define operations in terms of logic gates on qubits.



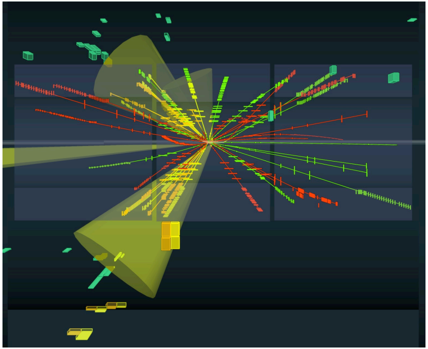
Why Quantum Computing for Particle Physics?

DOE QuantISED initiative:

A. Cosmos and Qubits: applications of quantum error correction, scrambling, computational complexity to black holes and holography
B. Foundational QIS-HEP Theory and Simulations: applications of quantum simulation; applications of quantum information and entanglement to field theory techniques
C. Quantum Computing for HEP: applications of quantum machine learning and data analysis tools
D. QIS-based Quantum Sensors: applications of quantum sensors to experiments
E. Research Technology for QIST: apply accelerator technology to qubit technology

Goals of This Work

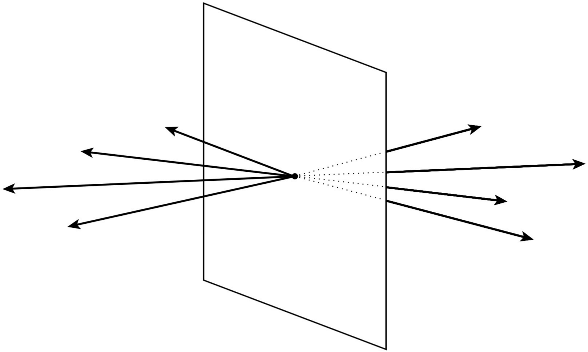
- Jets are collections of collimated, energetic hadrons formed in high-energy collisions.
- Identifying jets can be computationally intensive. Can quantum computers help?
- What would a realistic application of quantum search look like?
- Goal #1: investigate quantum algorithms. ✓
- Goal #2: improve classical algorithms. ✓



Case Study: Thrust

(Brandt, Peyrou, Sosnowski, Wroblewski '64; Farhi '77)

- For e^+e^- collision, find optimal separating plane to partition into two jets.



- Why thrust?
  - Thrust is a global optimization problem!
  - anti-k_t (default LHC algorithm) is $O(N \log N)$, vs $O(N^3)$ for thrust, but it is a local heuristic.

Quantum Computing

We will apply two key quantum computing paradigms.

- Quantum annealing: used in commercial hardware like D-Wave.
- Grover search: workhorse algorithm in quantum gate model.
  - Grover search finds desired item in size N database using only $O(\sqrt{N})$ time.
  - This is not realistic! Loading data that lives in classical memory takes at least $O(N)$ time.
  - To obtain speedup, size of search space needs to scale like $O(N^\alpha)$, with α offsetting data loading cost.

Results

Implementation	Time Usage	Qubit Usage
Classical (Yamamoto '84)	$O(N^3)$	—
Classical: Sort	$O(N^2 \log N)$	—
Classical: Parallel Sort	$O(N \log N)$	—
Quantum Annealing	Gap Dependent	$O(N)$
Quantum Search: Sequential	$O(N^2)$	$O(\log N)$
Quantum Search: Parallel	$O(N \log N)$	$O(N \log N)$

- We improve the classical thrust algorithm using a trick from (Salam, Soyez '07)
- We consider two quantum data loading models, **sequential** (fewer qubits but more time per query) and **parallel** (more qubits and less time per query).

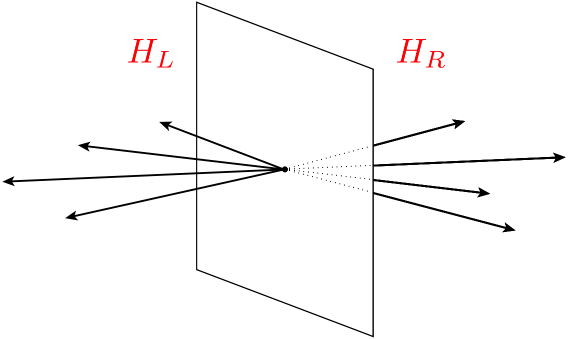
Outline

Definition of Thrust: Partitioning Problem

- Given N three-momenta $\{\vec{p}_i\}$ in CM frame, find partition $H_L \cup H_R$ maximizing

$$T(H_L) = \frac{2\left|\sum_{i\in H_L} \vec{p}_i\right|}{\sum_{i=1}^{N} |\vec{p}_i|} = \frac{2\left|\sum_{i\in H_R} \vec{p}_i\right|}{\sum_{i=1}^{N} |\vec{p}_i|}.$$

- Brute-force classical strategy: $O(N\,2^N)$. Motivates quantum annealing algorithm.

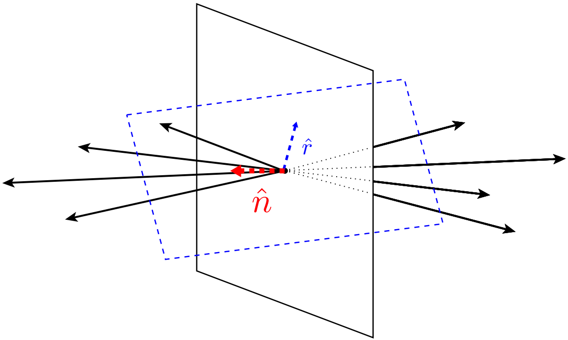


Definition of Thrust: Axis-Finding Problem

- For $\hat{n}$ a unit norm vector, find $\hat{n}_{opt}$ maximizing

$$T(\hat{n}) = \frac{\sum_{i=1}^{N} |\hat{n} \cdot \vec{p}_i|}{\sum_{i=1}^{N} |\vec{p}_i|} = \frac{2\sum_{i=1}^{N} \Theta(\hat{n} \cdot \vec{p}_i)(\hat{n} \cdot \vec{p}_i)}{\sum_{i=1}^{N} |\vec{p}_i|}.$$

- Partitions particles into $\hat{n}_{opt} \cdot \vec{p}_i > 0$ and $\hat{n}_{opt} \cdot \vec{p}_i < 0$.
- Motivates classical $O(N^3)$ algorithm and Grover search algorithm.



Definition of Thrust: Duality

(Thaler '15)

- These two definitions of thrust are in fact equivalent.
- Can be shown by directly optimizing, over partition H with $\vec{P} = \sum_{i \in H} \vec{p}_i$, the objective function

$$O(\vec{P}, \hat{n}) = \hat{n} \cdot \vec{P} + \lambda(\hat{n}^2 - 1).$$

  - Optimize over $\hat{n}$ first:

$$\hat{n}_{opt} = \frac{\vec{P}}{|\vec{P}|} \Rightarrow O(\vec{P}, \hat{n}_{opt}) = |\vec{P}|.$$

  - Optimize over $\vec{P}$ first:

$$\vec{P}_{opt} = \sum_{i=1}^{N} \Theta(\hat{n} \cdot \vec{p}_i)\vec{p}_i \Rightarrow O(\vec{P}_{opt}, \hat{n}) = \sum_{i=1}^{N} \Theta(\hat{n} \cdot \vec{p}_i)(\hat{n} \cdot \vec{p}_i).$$

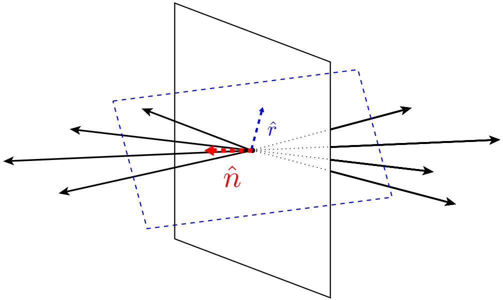
Outline

Classical Thrust Algorithm

(Yamamoto '84) Finds axis defining the separating plane.

1. For each pair $\vec{p_i}$, $\vec{p_j}$:
   - Compute reference vector $\hat{r}_{ij} = \frac{\vec{p_i} \times \vec{p_j}}{|\vec{p_i} \times \vec{p_j}|}$.
   - For each $\vec{p_k}$, assign to partition H_{ij} if $\hat{r}_{ij} \cdot \vec{p_k} > 0$, and ignore otherwise.
2. Return max $T(H_{ij})$.

- Requires time $O(N^3) = \underbrace{O(N^2)}_{\text{Number of } H_{ij}} \times \underbrace{O(N)}_{\text{Sum the } \vec{p_k} \text{ for each } H_{ij}}$.



Classical Thrust Algorithm: Improvements via Sort

(Using Salam, Soyez '07)

- Traverse particles in special order to avoid redundant $O(N)$ cost for each H_{ij}.

1. For each $\vec{p_i}$,
   - Sort $\vec{p_j}$ by azimuth b/w $\vec{p_j}$ and $\vec{p_i}$, obtaining $\{\vec{p_{j_1}}, \vec{p_{j_2}}, \dots\}$.
   - For $\vec{p_{j_1}}$, loop over all $\vec{p_k}$, assigning to partition H_{ij_1} if $\hat{r}_{ij_1} \cdot \vec{p_k} > 0$.
   - For $\vec{p_{j_n}}$, $n \geq 2$, obtain $T(H_{ij_n})$ from $T(H_{ij_{n-1}})$ using the fact that only one particle ever enters and leaves $H_{ij_{n-1}}$.

- Requires time

$$O(N^2 \log N) = \underbrace{O(N)}_{\text{Number of } \vec{p_i}} \times \Big[\underbrace{O(N \log N)}_{\text{Sort}} + \underbrace{O(N)}_{\text{Compute } T(H_{ij_1})} + \underbrace{O(N)}_{\text{Compute } T(H_{ij_n}), n\geq 2}\Big].$$

Outline

Quantum Annealing

- Existing hardware like D-Wave are quantum annealers.
- Hardware consists of spins on a lattice; challenge is controlling connectivity between spins.
- Solution to optimization problem is encoded in the ground state of an Ising-like Hamiltonian.

Quantum Annealing: Thrust (1/2)

- Equivalently, for $x_i \in \{0, 1\}$, we want to minimize the QUBO objective function

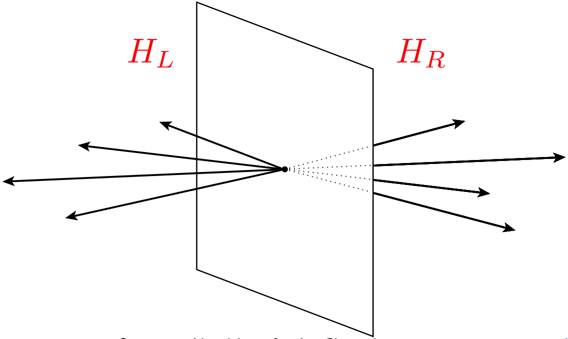
$$O(\{x_i\}) = \sum_{i,j=1}^{N} Q_{ij} x_i x_j .$$

- Remarkably, thrust can be written in QUBO form!
- Recall definition of thrust as a partitioning problem.

Quantum Annealing: Thrust (2/2)

- Partition H has total three-momentum $\vec{P}(\{x_i\}) = \sum_{i=1}^{N} \vec{p}_i x_i$, where $x_i = 1$ for $i \in H$ and $x_i = 0$ otherwise. Then $T(H) \propto |\vec{P}|$.
- This is monotonic $\Rightarrow$ square to obtain QUBO function:

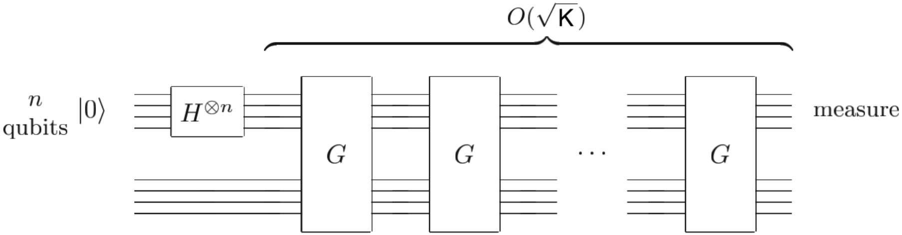
$$O(\vec{x}_i) = -|\vec{P}|^2 = -\sum_{i,j=1}^{N} (\vec{p}_i \cdot \vec{p}_j) x_i x_j .$$



Outline

Quantum Search (1/2)

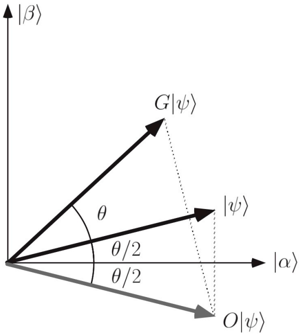
- Search is the backbone of optimization algorithms.
- Grover search performs search in the quantum gate model.
  - In this model, algorithm = prepare initial state, apply unitary gates, measure to obtain desired final state.
  - Grover uses $\sqrt{K}$ gates to search K items.



Quantum Search (2/2)

Grover search (Grover '96; Boyer, Brassard, Hoyer, Tapp '98):

1. Prepare initial state $|\psi_0\rangle = \frac{1}{\sqrt{K}} \sum_{i=1}^{K} |i\rangle$.
2. Repeat $O(\sqrt{K})$ times:
   - Reflect about marked states.
   - Reflect about initial state $|\psi_0\rangle$.



Quantum Search: Maximum Finding

- Maximum finding algorithm (Durr, Hoyer '96): Return max of array using Grover search.
  - Track best max so far; marked states are those larger than current max.
- We will apply this algorithm to thrust search space of size $K = O(N^2) \Rightarrow$ outer Grover loop scales like $O(N)$.

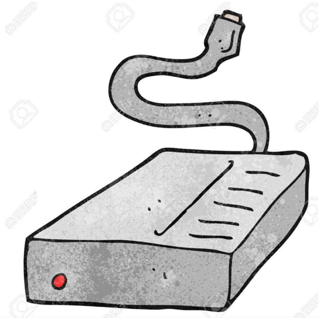
Quantum Search: Thrust Algorithm (1/2)

Schematically,

1. Pick random indices m, n corresponding to current max.
2. Repeat $O(N)$ times:
   - Prepare initial state $|\psi_0\rangle = \frac{1}{2N}\sum_{i,j=1}^{2N} |i\rangle |j\rangle |0\rangle |0\rangle |\hat{0}\rangle |0\rangle |0\rangle$.
   - Reflect about marked states:
     2.1 Compute T_{ij}, mapping $|i\rangle |j\rangle |0\rangle |0\rangle |\hat{0}\rangle |0\rangle |0\rangle \mapsto |i\rangle |j\rangle |0\rangle |0\rangle |\hat{0}\rangle |0\rangle |T_{ij}\rangle$.
     2.2 Apply phase to states with $T_{ij} > T_{mn}$, mapping $|i\rangle |j\rangle |0\rangle |0\rangle |\hat{0}\rangle |0\rangle |T_{ij}\rangle \mapsto (-1)^{\Theta(T_{ij}-T_{mn})} |i\rangle |j\rangle |0\rangle |0\rangle |\hat{0}\rangle |0\rangle |T_{ij}\rangle$.
     2.3 Uncompute intermediate registers.
   - Reflect about initial state.
   - Measure i, j register. If $T_{ij} > T_{mn}$ update m, n.

Quantum Search: Data Loading (1/2)

- Grover algorithm assumed ability to prepare superposition over all items.
- Collider data lives on classical hard drive, not in quantum superposition!



Quantum Search: Data Loading (2/2)

- Data loading needs to happen in the step that computes T_{ij}.
- Define two abstract unitary operations:
  - LOOKUP: $U_{LOOKUP}\left|i\right\rangle\left|\vec{0}\right\rangle = \left|i\right\rangle\left|\vec{p_i}\right\rangle$.
  - SUM: $U_{SUM}\left|c\right\rangle\left|0\right\rangle = \left|c\right\rangle\left|\sum_{k=1}^{N} f(\vec{p_k}; c)\right\rangle$.
- Essentially, the data on the hard drive tell us which unitaries to apply.

Quantum Search: Thrust Algorithm (2/2)

How to compute T_{ij} using LOOKUP and SUM:

1. Load $\vec{p_i}, \vec{p_j}$ using LOOKUP: $|i\rangle\,|j\rangle\,|\vec{0}\rangle\,|\vec{0}\rangle\,|\hat{0}\rangle\,|\vec{0}\rangle\,|0\rangle \mapsto |i\rangle\,|j\rangle\,|\vec{p_i}\rangle\,|\vec{p_j}\rangle\,|\hat{0}\rangle\,|\vec{0}\rangle\,|0\rangle$.
2. Calculate $\hat{r_{ij}}$ using $\vec{p_i}$, $\vec{p_j}$: $|i\rangle\,|j\rangle\,|\vec{p_i}\rangle\,|\vec{p_j}\rangle\,|\hat{0}\rangle\,|\vec{0}\rangle\,|0\rangle \mapsto |i\rangle\,|j\rangle\,|\vec{p_i}\rangle\,|\vec{p_j}\rangle\,|\hat{r_{ij}}\rangle\,|\vec{0}\rangle\,|0\rangle$.
3. Apply SUM with $f(\vec{p_k};\hat{r_{ij}}) = \{\vec{p_k} \text{ if } \hat{r_{ij}}\cdot\vec{p_k} > 0; \vec{0} \text{ if } \hat{r_{ij}}\cdot\vec{p_k} < 0\}$ to sum momentum in H_{ij}: $|i\rangle\,|j\rangle\,|\vec{p_i}\rangle\,|\vec{p_j}\rangle\,|\hat{r_{ij}}\rangle\,|\vec{0}\rangle\,|0\rangle \Rightarrow |i\rangle\,|j\rangle\,|\vec{p_i}\rangle\,|\vec{p_j}\rangle\,|\hat{r_{ij}}\rangle\,|\vec{P_{ij}}\rangle\,|0\rangle$.
4. Divide by normalization factors to obtain thrust: $|i\rangle\,|j\rangle\,|\vec{p_i}\rangle\,|\vec{p_j}\rangle\,|\hat{r_{ij}}\rangle\,|\vec{P_{ij}}\rangle\,|0\rangle \mapsto |i\rangle\,|j\rangle\,|\vec{p_i}\rangle\,|\vec{p_j}\rangle\,|\hat{r_{ij}}\rangle\,|\vec{P_{ij}}\rangle\,|T_{ij}\rangle$.
5. Uncompute intermediate registers: $|i\rangle\,|j\rangle\,|\vec{p_i}\rangle\,|\vec{p_j}\rangle\,|\hat{r_{ij}}\rangle\,|\vec{P_{ij}}\rangle\,|T_{ij}\rangle \mapsto |i\rangle\,|j\rangle\,|\vec{0}\rangle\,|\vec{0}\rangle\,|\hat{0}\rangle\,|\vec{0}\rangle\,|T_{ij}\rangle$.

Runtime: $O(N(C_{LOOKUP} + C_{SUM}))$.

Quantum Search: Sequential Model

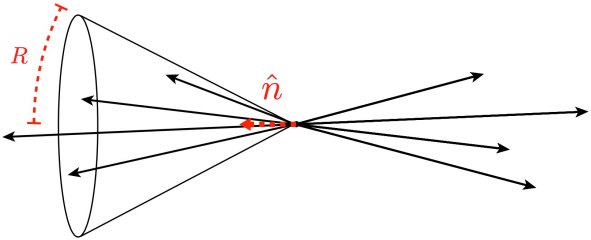
- LOOKUP and SUM require $O(N)$ time and $O(\log N)$ qubits.
- LOOKUP has one register of size $O(\log N)$ to store i, $\vec{p}_i$.
- LOOKUP scans all N items in classical database, returning $\vec{p}_i$.
- SUM scans all N items in classical database to compute $\sum_{i=1}^{N} f(\vec{p}_i; c)$.
- Thrust algorithm requires $O(N^2)$ time and $O(\log N)$ qubits.

Outline

Towards Jet Algorithms for the LHC

How does the thrust case differ from LHC jet algorithms?

- Different coordinate system from e^+e^- collisions.
- Divide event into single jet with opening angle R, plus unclustered region:



- Thrust case is $R = \pi/2$.

Jet Function Maximization

(Georgi '14)

- Find the subset of particles that maximizes the following function ($\beta = 1/(1 - \cos R)$):

$$O(P_\mu) = E - \beta \frac{M^2}{E}.$$

- This is a global optimization problem!
- Quantum anneal using modified QUBO objective function:

$$O'(P_\mu) = E^2 - \beta M^2 = \sum_{i,j=1}^{N} \beta \left(\vec{p_i} \cdot \vec{p_j} + \frac{1-\beta}{\beta} E_i E_j \right) x_i x_j.$$

Multi-region Optimization

Simultaneously find M jets.

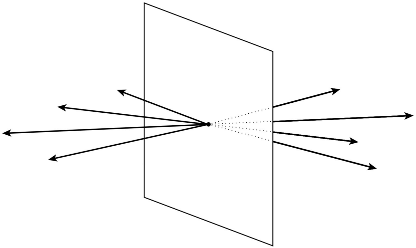
- $O(N(M+1))$ qubits x_{ir}, $i \in \{1, 2, ..., N\}$, $r \in \{1, 2, ..., M\}$.
- Generalize modified jet function objective function

$$O(\{x_{ij}\}) = \sum_{r=1}^{M} \sum_{i,j=1}^{N} \beta \left(\vec{p_i} \cdot \vec{p_j} + \frac{1-\beta}{\beta} E_i E_j \right) x_{ir} x_{jr} + \Lambda \sum_{i=1}^{N} \left(1 - \sum_{r=0}^{M} x_{ir} \right)^2 .$$

- Last penalty term ensures each particle is only assigned to one region.

Outline

Conclusions



- Explored how to realistically apply quantum computing to collider physics problems.
  - In particular, how do we handle data loading in the quantum gate model?
  - Improved classical $O(N^3)$ thrust algorithm to $O(N^2 \log N)$ classical algorithm and $O(N^2)$ quantum algorithm.
- Considerations are relevant to broader optimization and clustering problems.

Future Directions

- Can we identify other interesting problems from collider physics where the cost of data read-in is $O(N)$, while the size of the search space goes like $O(N^\alpha)$ for large α?