# Quantum Algorithms for Collider Physics

Jesse Thaler, MIT

Case Western
Nov 15, 2019

based on 1908.08949

Annie Wei, Preksha Naik
Aram Harrow, JDT

---

Collider data analysis is based on a variety of algorithms, and computational complexity is one metric via which we evaluate the feasibility of different data analysis strategies,

e.g. anti-$k_T$ : IRC safe (theory)

~Circular jets (expt)

$\mathcal{O}(N \log N)$ (computation)

Obviously, we use classical computational complexity at the moment, but the rise of quantum computers raises the question about quantum complexity.

Can we (in principle) speed up collider algorithms using a quantum computer?

Today: Case study of "thrust".

Punchline: Data loading is a major bottleneck for quantum algorithms that take in classical inputs.

$N$ inputs $\Rightarrow$ $\mathcal{O}(N)$ runtime for loading

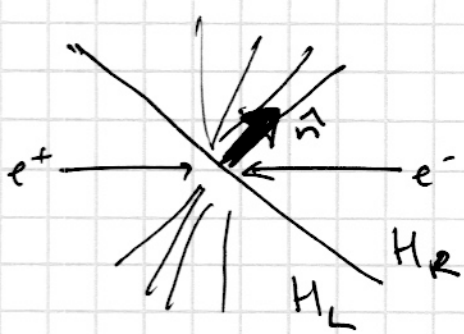
This limits achievable gains.

Lessons: Quantum-Inspired Classical Algorithms are an important by-product of this kind of research.

---

Review of Thrust

$e^+e^- \to$ hadrons, partition event into hemispheres.




History going back to Brandt, Peyron, Sosnowski, Wroblewski (1964)

Made famous by Farhi (1977)

Scrutinized by De Rujula, Ellis, Floratos, Gaillard (1978)

Brandt Dahmen (1979)

Fastest known algorithm: Yamamoto (1983)

Has two dual formulations that are easiest to see ~~by~~ thinking about a meta problem:

$$T(\vec{p}, \hat{n}) = \hat{n}\cdot\vec{p} + \lambda(\hat{n}^2 - 1)$$

$\vec{p}$ — momentum for a partition; $\hat{n}$ — axis; $\lambda$ — Lagrange multiplier

$$\vec{p} = \sum_{i\in H} \vec{p}_i = \sum_{i=1}^{N} \vec{p}_i\, x_i$$

$x_i$: 0 out, 1 in.

Fix $\vec{p}$: $\hat{n}_{opt} = \dfrac{\vec{p}}{|\vec{p}|}$ (thrust axis aligns with momentum in a partition)

$$\Rightarrow T(\vec{p}) = |\vec{p}| \quad \text{(goal to maximize 3-momentum)}$$

Fix $\hat{n}$: $\vec{p}_{opt} = \sum_{i=1}^{N} \Theta(\hat{n}\cdot\vec{p}_i)\,\vec{p}_i$ (partition into hemisphere)

$$\Rightarrow T(\hat{n}) = \sum_{i=1}^{N} \Theta(\hat{n}\cdot\vec{p}_i)\,\hat{n}\cdot\vec{p}_i \quad \text{(find axis whose hemisphere has maximum overlap.)}$$

$$\text{Partitioning Problem} \xleftrightarrow{\text{dual}} \text{Axis Finding}$$

Partitioning Problem: naturally suited for quantum annealing

$$\text{Axis Finding} \overset{1983}{\Downarrow} \text{Reference axis Finding}$$

Reference axis Finding: naturally suited for Grover search.

Quantum Annealing:

e.g. D-Wave device, can find ground state to:

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

$\uparrow$ 0 or 1

Specific implementation on flux qubits (kind of like a bunch of connected SQUIDs)

Thrust problem SQUARED is "QUBO" problem

Quadratic Unconstrained Binary Optimization

$$T(\{x_i\})^2 = \frac{4}{\text{normalization}} \sum_{i,j=1}^{N} \vec{p}_i \cdot \vec{p}_j\, x_i x_j$$

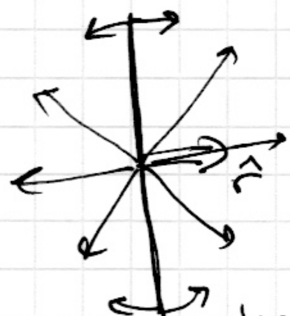
No provable bounds on quantum advantage. Depends on gap.

Grover Search.

If you have $K$ objects to search over, can do it in $\mathcal{O}(\sqrt{K})$ time on universal quantum computer.

Thrust is a reference axis search problem.

$\hat{n}$ = thrust axis $\qquad$ $\hat{r}$ = reference axis with same partition as thrust





Extreme case: partition wall on top of two particles!

$$\hat{r}_{ij} = \frac{\vec{p}_i \times \vec{p}_j}{|\vec{p}_i \times \vec{p}_j|}$$ ← plane normal to this contains $\vec{p}_i$ and $\vec{p}_j$

Best algorithm

$$\mathcal{O}(N^3) = \underbrace{\mathcal{O}(N^2)}_{\text{partitions}} + \underbrace{\mathcal{O}(N)}_{\text{determine } |\vec{P}|.}$$

How does Grover work?

initialize: $|\psi_0\rangle = \frac{1}{\sqrt{K}} \sum_{i=1}^{K} |i\rangle |0\rangle$

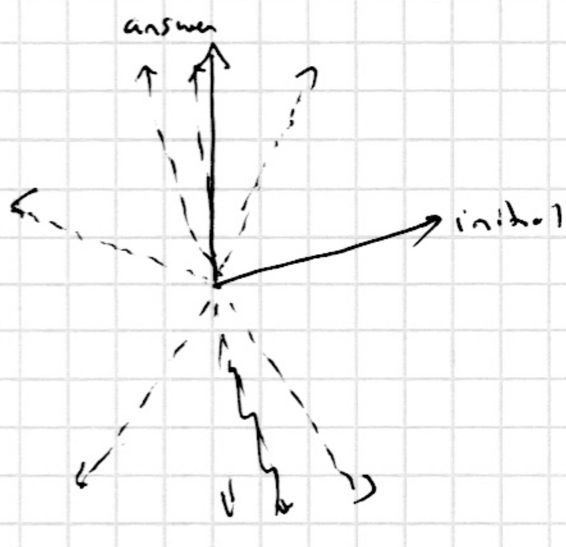
mark. $\Rightarrow \frac{1}{\sqrt{K}} \sum_{i=1}^{K} |i\rangle |f_i\rangle$

flip $\Rightarrow \frac{1}{\sqrt{K}} \sum_{i=1}^{k} (-1)^{f_i} |i\rangle |f_i\rangle$

uncompute $\Rightarrow \frac{1}{\sqrt{K}} \sum_{i=1}^{K} (-1)^{f_i} |i\rangle |0\rangle$

reflect $\Rightarrow \left(2|\psi_0\rangle\langle\psi_0| - \mathbb{I}\right) |\text{state}\rangle$

repeat for $\sqrt{K}$ steps.




(Hopefully this works on the blackboard.)

For thrust we want maximum. so.

$$|ij\rangle|0\rangle \cancel{|\text{max so far}\rangle}$$

start classical $\downarrow$ $\Big\downarrow$ $O(N)$ because you have to compute $T_{ij}$!

$$|ij\rangle|T_{ij}\rangle \cancel{|\text{max so far}\rangle}$$

$$(-1)^{\Theta(T_{ij} > T_{\text{max so far}})} \; |ij\rangle|T_{ij}\rangle \cancel{|\text{max so far}\rangle}$$

$$(-1)^{\Theta} \text{ - - - } \; |ij\rangle|0\rangle \cancel{|\text{max so far}\rangle}$$

You get closer to a state with bigger thrust.

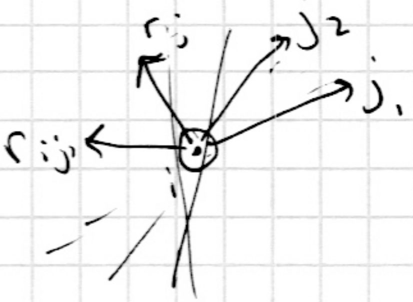
Final algorithm

$$O(N^2) = \underbrace{O\left(\sqrt{N^2}\right)}_{\text{Grover max finding}} \times \underbrace{O(N)}_{\text{data loading is key bottleneck}}$$

Coda: We found $N^2 \log N$ classical algorithm, inspired by our research. 8

Based on Salam Soyez 2007

$\hat{r}_{ij} \leftarrow O(N^2)$ but you can save yourself some computing time




One partition input at a time!

$$\underbrace{O(N)}_{\text{choices for } i} \times \left( \underbrace{O(N \log N)}_{\text{sorting } j} + \underbrace{O(N)}_{\text{compute thrust for first object}} + \underbrace{N \cdot O(1)}_{\text{update that going around.}} \right)$$

Final Result: $O(N^3)$ previous classical

$O(N^2 \log N)$ sorting classical

$\approx O(N^2)$ best quantum...

We found an improved algorithm, both classical and quantum, speed comes from completely different source, (sort vs. Grover)

Broader perspective:

I no longer think we'll get a dramatic speed up using quantum algorithms for individual jets.

But still work to be done on ensembles!

$$N_{\text{events}/\text{run}} \gg N_{\text{particles}/\text{event}}$$

Histogramming : $\mathcal{O}(N_{\text{events}/\text{run}})$

We don't even talk about $\mathcal{O}(N^2)$!

But as you ~~[illegible]~~ saw in my colloquium, interesting $\mathcal{O}(N^2)$ problems!

Can we speed this up to $\mathcal{O}(N \log N)$ on quantum device (or quantum-inspired algorithm)?

What other quantum tricks can we leverage?