Reconstructing Proton-Proton Collision Positions at the Large Hadron Collider with D-Wave

Newport, RI
September 26, 2018

Andrew J. Wildridge  Sachin B. Vaidya  Andreas Jung  Souvik Das
Outline

- Introduction to the Large Hadron Collider and the Compact Muon Solenoid detector
- QUBO formulation of the problem for D-Wave
- D-Wave performance, benchmarked against CPU
- Conclusions and outlook
The Large Hadron Collider

- The Higgs boson
- Large extra-dimensions
- Supersymmetry
- Dark matter
- Baryogenesis
Looking for: A high-energy physics problem that has a natural formulation for quantum annealing, and is simple
Proton-proton collisions at the Large Hadron Collider

Looking for: A high-energy physics problem that has a natural formulation for quantum annealing, and is simple

Chosen problem: Reconstructing proton-proton collision positions at the Large Hadron Collider (LHC)

- The LHC circulates protons inside its beam-pipes not in a continuous stream but in several closely packed bunches.
- Each bunch contains ~ 100 billion protons
When counter-rotating bunches cross, only ~ 20 protons collide in one straight line.

Each p-p collision results in ~ 50 “interesting” tracks from charged particles produced.

**Looking for:** A high-energy physics problem that has a natural formulation for quantum annealing, and is simple.

**Chosen problem: Reconstructing proton-proton collision positions at the Large Hadron Collider (LHC)**

- The LHC circulates protons inside its beam-pipes not in a continuous stream but in several closely packed bunches.
- Each bunch contains ~ 100 billion protons.
- When counter-rotating bunches cross, only ~ 20 protons collide in one straight line.
- Each p-p collision results in ~ 50 “interesting” tracks from charged particles produced.
When counter-rotating bunches cross, only ~ 20 protons collide in one straight line. Each p-p collision results in ~ 50 “interesting” tracks from charged particles produced.

Looking for: A high-energy physics problem that has a natural formulation for quantum annealing, and is simple.

**Chosen problem:** Reconstructing proton-proton collision positions at the Large Hadron Collider (LHC)

- The LHC circulates protons inside its beam-pipes not in a continuous stream but in several closely packed bunches.
- Each bunch contains ~ 100 billion protons.
- When counter-rotating bunches cross, only ~ 20 protons collide in one straight line.
- Each p-p collision results in ~ 50 “interesting” tracks from charged particles produced.

**Which tracks come from which p-p collision?**

**Where are the p-p collision points in a bunch?**
The CMS detector observes particles created at LHC collisions.

- The Compact Muon Solenoid is a particle detector at one (of four) p-p crossing point at the LHC.
- Charged particles are reconstructed as tracks. All reconstructions come with uncertainties.
The CMS detector observes particles created at LHC collisions:

- The Compact Muon Solenoid is a particle detector at one (of four) p-p crossing point at the LHC.
- Charged particles are reconstructed as tracks. All reconstructions come with uncertainties.
- Where a track approaches the beam (z-)axis closest has uncertainties $z_i \pm \delta z_i$.
- Uncertainties obscure which tracks originated together at p-p collision.

In the x-y plane, Particle trajectories reconstructed as tracks. Reconstructions come with uncertainties:

\[
\text{Tracks extrapolated to \textendash\textquotedblleft cut\textendash\textquotedblright beam axis } z_i \text{ \ at positions in } z_i.
\]

In the y-z plane, Particle trajectories reconstructed as tracks. Reconstructions come with uncertainties:

\[
\text{Tracks extrapolated to \textendash\textquotedblleft cut\textendash\textquotedblright beam axis } z_i \text{ \ at positions in } z_i.
\]
The CMS detector observes particles created at LHC collisions.

- The Compact Muon Solenoid is a particle detector at one (of four) p-p crossing point at the LHC.
- Charged particles are reconstructed as tracks. All reconstructions come with uncertainties.
- Where a track approaches the beam (z-)axis closest has uncertainties $z_i \pm \delta z_i$.
- Uncertainties obscure which tracks originated together at p-p collision.
- Position of p-p collisions reduced to a clustering problem in 1-D.
- Solved in CMS using Deterministic Annealing. Called “Primary Vertexing.”
The CMS detector observes particles created at LHC collisions. The Compact Muon Solenoid is a particle detector at one (of four) p-p crossing point at the LHC. Charged particles are reconstructed as tracks. All reconstructions come with uncertainties.

- Where a track approaches the beam (z-) axis closest has uncertainties $z_i \pm \delta z_i$
- Uncertainties obscure which tracks originated together at p-p collision
- Position of p-p collisions reduced to a clustering problem in 1-D
- Solved in CMS using Deterministic Annealing. Called “Primary Vertexing”

Can D-Wave solve it using quantum annealing?
Track clustering QUBO formulation for D-Wave

\[ H_p = \sum_{k} \sum_{i} \sum_{j>i} n_T p_{ik} p_{jk} g(D(i, j); m) \]

\[ + \lambda \sum_{i} \left( 1 - \sum_{k} p_{ik} \right)^2, \]

* Clustering problem naturally expressed in QUBO form (V. Kumar, et. al. “Quantum annealing for combinatorial clustering” Quantum Inf. Processing 17 (2018) 39)
Track clustering QUBO formulation for D-Wave

\[
H_p = \sum_k \sum_i \sum_{j>i} p_{ik} p_{jk} g(D(i, j); m) + \lambda \sum_i \left(1 - \sum_k p_{ik}\right)^2,
\]

- Probability (0 or 1) of \(i\)th track to have come from \(k\)th p-p collision is \(p_{ik}\). Element \(p_{ik}\) is represented by a qubit
Track clustering QUBO formulation for D-Wave

  - Probability (0 or 1) of \( i \)-th track to have come from \( k \)-th p-p collision is \( p_{ik} \). Element \( p_{ik} \) is represented by a qubit
  - Coupling between two qubits \( p_{ik} \) and \( p_{jk} \) that represent association of two tracks to same p-p collision \( k \) is a distance measure between the tracks \( D(i, j) \). Punish associations corresponding to widely separated tracks
  - \( D(i, j) \) is Manhattan distance attenuated by uncertainty

\[
D(i, j) = \frac{|z_i - z_j|}{\sqrt{\delta z_i^2 + \delta z_j^2}}.
\]

\[
H_p = \sum_{k} \sum_{i} \sum_{j > i} p_{ik} p_{jk} g(D(i, j); m) + \lambda \sum_{i} \left(1 - \sum_{k} p_{ik}\right)^2,
\]
Track clustering QUBO formulation for D-Wave

Clustering problem naturally expressed in QUBO form \((V.\text{Kumar, et. al.}\) “Quantum annealing for combinatorial clustering” Quantum Inf. Processing 17 (2018) 39)

- Probability (0 or 1) of 1th track to have come from 1th p-p collision is \(p_k\). Element \(p_k\) **is represented by a qubit**
- Coupling between two qubits \(p_{ik}\) and \(p_{jk}\) that represent association of two tracks to same p-p collision \(k\) is a distance measure between the tracks \(D(i, j)\). **Punish associations corresponding to widely separated tracks**
  - \(D(i, j)\) is Manhattan distance attenuated by uncertainty \(D(i, j) = \frac{|z_i - z_j|}{\sqrt{\delta z_i^2 + \delta z_j^2}}\).
  - \(g(D(i, j); m)\) seeks to distribute the couplings evenly without changing order. Else, lots of small couplings and some large couplings for p-p finding problem. Not for all clustering problems. Empirically seen to improve results with \(m = 5\)

\[ g(x; m) = 1 - e^{-mx} \]
Track clustering QUBO formulation for D-Wave

- Clustering problem naturally expressed in QUBO form (*V. Kumar, et. al.* “Quantum annealing for combinatorial clustering” Quantum Inf. Processing 17 (2018) 39)

  - Probability (0 or 1) of \(i\)th track to have come from \(k\)th p-p collision is \(p_k\). Element \(p_k\) is represented by a qubit
  - Coupling between two qubits \(p_k\) and \(p_k\) that represent association of two tracks to same p-p collision \(k\) is a distance measure between the tracks \(D(i, j)\). Punish associations corresponding to widely separated tracks
    - \(D(i, j)\) is Manhattan distance attenuated by uncertainty \(D(i, j) = \frac{|z_i - z_j|}{\sqrt{\delta z_i^2 + \delta z_j^2}}\).
    - \(g(D(i, j); m)\) seeks to distribute the couplings evenly without changing order. Else, lots of small couplings and some large couplings for p-p finding problem. Not for all clustering problems. Empirically seen to improve results with \(m = 5\)
      \[
g(x; m) = 1 - e^{-mx},
\]
    - Bias per qubit comes from \(\lambda\) term enforcing one track associated with one p-p collision. \(\lambda = 1.2 \max(D(i, j))\) is optimal

\[
H_p = \sum_k \sum_i \sum_{j>i} p_ip_jg(D(i, j); m) + \lambda \sum_i \left(1 - \sum_k p_k\right)^2,
\]

![Diagram of p-p collision number and tracks cutting beam axis](Image)
Track clustering QUBO formulation for D-Wave

- Clustering problem naturally expressed in QUBO form (V. Kumar, et. al. “Quantum annealing for combinatorial clustering” Quantum Inf. Processing 17 (2018) 39)

- Probability (0 or 1) of \( i^{th} \) track to have come from \( k^{th} \) p-p collision is \( p_k \). Element \( p_k \) is represented by a qubit.

- Coupling between two qubits \( p_k \) and \( p_k \) that represent association of two tracks to same p-p collision \( k \) is a distance measure between the tracks \( D(i, j) \). Punish associations corresponding to widely separated tracks.

- \( D(i, j) \) is Manhattan distance attenuated by uncertainty \( D(i, j) = \frac{|z_i - z_j|}{\sqrt{\delta z_i^2 + \delta z_j^2}} \).

- \( g(D(i, j); m) \) seeks to distribute the couplings evenly without changing order. Else, lots of small couplings and some large couplings for p-p finding problem. Not for all clustering problems. Empirically seen to improve results with \( m = 5 \).

- Bias per qubit comes from \( \lambda \) term enforcing one track associated with one p-p collision. \( \lambda = 1.2 \max(D(i, j)) \) is optimal.

We use D-Wave’s default embedding algorithms.
Performance on one event with 3 p-p collision, 15 tracks
Generating artificial events

- Algorithm tested on artificial events drawn from measured LHC distributions of collision points and measured CMS distributions of tracks
- Realistic track reconstruction uncertainties used. CMS Collaboration, JINST 9 (2014) P10009
Performance on one event with 3 p-p collision, 15 tracks

Generating artificial events

- Algorithm tested on artificial events drawn from measured LHC distributions of collision points and measured CMS distributions of tracks
- Realistic track reconstruction uncertainties used. CMS Collaboration, JINST 9 (2014) P10009
Performance on one event with 3 p-p collisions and 15 tracks

- QUBO bias terms are equal and come from the λ constraint
- Quantum state prepared and annealed 10,000 times. **DW_2000Q_2_1** used
  - 6,825 solutions are valid, i.e. λ constraint is strictly met (∑_k ρ_k = 1 for all tracks)
  - 6,615 solutions are correct (Solution 1). **Convergence efficiency = 66%**
  - Small number of valid secondary solutions where one track has been misassociated

Generating artificial events

- Algorithm tested on artificial events drawn from **measured LHC distributions of collision points** and **measured CMS distributions of tracks**
- **Realistic track reconstruction uncertainties** used. *CMS Collaboration, JINST 9 (2014) P10009*

Coeficients of QUBO form used to solve particular event with 3 collisions, 15 tracks

Energy spectrum of solutions for one event with 3 p-p collisions and 15 tracks explored by the D-Wave 2000Q_2_1 with 10,000 samples. Energies corresponding to valid solutions, where ρ_k add up to 1 for every track, are plotted with solid lines while invalid solutions are plotted with dashed lines. Error bars correspond to statistical uncertainties. The best and next-to-best valid solutions are indicated as Solutions 1 and 2, respectively. For clarity, the histogram is binned by 1 GHz below 10 GHz, by 10 GHz for 10 — 100 GHz, and by 100 GHz above 100 GHz
Benchmarking against simulated annealing in equal time

Principle: Equalize working time between CPU and QPU, and compare convergence efficiency
Benchmarking against simulated annealing in equal time

Principle: Equalize working time between CPU and QPU, and compare convergence efficiency

Sampling time of D-Wave 2000Q_2_1

- Total sample time = 164 μs
  - Anneal time = 20 μs
  - Readout time = 123 μs
  - Delay time = 21 μs
- How many Simulated Annealing sweeps can we fit in this?
  - Depends on problem complexity
Benchmarking against simulated annealing in equal time

Principle: Equalize working time between CPU and QPU, and compare convergence efficiency

**Sampling time of D-Wave 2000Q_2_1**
- Total sample time = 164 μs
- Anneal time = 20 μs
- Readout time = 123 μs
- Delay time = 21 μs
- How many Simulated Annealing sweeps can we fit in this?
  - Depends on problem complexity

**CPU:** 3.1 GHz Intel Core i7-5557U (MacBook pro 2017)
**Algorithm:** Simulated annealing. Time optimized
- Use a sorted `std::map` with keys = bit index, value = list of other bits it couples to and the coupling
- Bit flip only requires to compute energy difference

**Compiler:** C++, -O2 optimization
Benchmarking against simulated annealing in equal time

Principle: Equalize working time between CPU and QPU, and compare convergence efficiency

Sampling time of D-Wave 2000Q_2_1
- Total sample time = 164 µs
  - Anneal time = 20 µs
  - Readout time = 123 µs
  - Delay time = 21 µs
- How many Simulated Annealing sweeps can we fit in this?
  - Depends on problem complexity

CPU: 3.1 GHz Intel Core i7-5557U (MacBook pro 2017)
Algorithm: Simulated annealing. Time optimized
  - Use a sorted std::map with keys = bit index, value = list of other bits it couples to and the coupling
  - Bit flip only requires to compute energy difference
Compiler: C++, -O2 optimization

CPU process time against number of sweeps for various event topologies under consideration
- 2 vertices, 10 tracks: 3.9 µs/sweep
- 3 vertices, 9 tracks: 5.3 µs/sweep
- 2 vertices, 16 tracks: 7.8 µs/sweep
- 3 vertices, 15 tracks: 10.9 µs/sweep
- 4 vertices, 12 tracks: 10.5 µs/sweep
- 4 vertices, 15 tracks: 16.5 µs/sweep
- 5 vertices, 15 tracks: 19.7 µs/sweep

Estimating CPU time per sweep
- Measure process time, not wall time
- Plot process time against nSweeps for different event topologies
- Discard overhead. Slope is time per sweep.
  - For 3 collision 15 tracks, 10.9 µs/sweep. Thus, 15 sweeps would fit in D-Wave’s sampling time
Benchmarking against simulated annealing in equal time

Principle: Equalize working time between CPU and QPU, and compare convergence efficiency

**Sampling time of D-Wave 2000Q_2_1**
- Total sample time = 164 μs
  - Anneal time = 20 μs
  - Readout time = 123 μs
  - Delay time = 21 μs
- How many Simulated Annealing sweeps can we fit in this?
  - Depends on problem complexity

Simulated annealing on CPU is only allowed as many iterations between $\beta_{\text{init}} = 0.1$ and $\beta_{\text{final}} = 10$ as would fit 164 μs

**CPU:** 3.1 GHz Intel Core i7-5557U (MacBook pro 2017)
**Algorithm:** Simulated annealing. Time optimized
- Use a sorted `std::map` with keys = bit index, value = list of other bits it couples to and the coupling
- Bit flip only requires to compute energy difference

**Compiler:** C++, -O2 optimization

Estimating CPU time per sweep
- Measure process time, not wall time
- Plot process time against nSweeps for different event topologies
- Discard overhead. **Slope** is time per sweep.
  - For 3 collision 15 tracks, 10.9 μs/sweep. Thus, 15 sweeps would fit in D-Wave’s sampling time
Performance on 100 events with 3 p-p collisions and 15 tracks

- 100 events with 3 p-p collisions and 15 tracks are thrown from measured CMS distributions
  - Each event is sampled 10,000 times by both the QPU and the CPU (in equivalent time)
  - Events with collisions spread closely compared to the spread of their tracks are hard for both QPU and CPU to solve
  - A distribution of convergence efficiencies is observed
    - QPU: mean = 42%, std. dev. = 25%
    - CPU: mean = 24%, std. dev. = 11%
Performance on 100 events with 3 p-p collisions and 15 tracks

Performance on an ensemble of events

- 100 events with 3 p-p collisions and 15 tracks are thrown from measured CMS distributions
- Each event is sampled 10,000 times by both the QPU and the CPU (in equivalent time)
- Events with collisions spread closely compared to the spread of their tracks are hard for both QPU and CPU to solve
- A distribution of convergence efficiencies is observed
  - QPU: mean = 42%, std. dev. = 25%
  - CPU: mean = 24%, std. dev. = 11%

Is there underlying structure to these distributions?
Performance on 100 events with 3 p-p collisions and 15 tracks

- 100 events with 3 p-p collisions and 15 tracks are thrown from measured CMS distributions
  - Each event is sampled 10,000 times by both the QPU and the CPU (in equivalent time)
  - Events with collisions spread closely compared to the spread of their tracks are hard for both QPU and CPU to solve
  - A distribution of convergence efficiencies is observed
    - QPU: mean = 42%, std. dev. = 25%
    - CPU: mean = 24%, std. dev. = 11%

Is there underlying structure to these distributions?

Performance against event “clumpiness”

- A measure of event clumpiness is the Dunn index. Low Dunn index = diffuse event, high = clumpy event
  
  \[ D = \frac{\min_{1 \leq i < j \leq n} d(i, j)}{\max_{1 \leq k \leq n} d'(k)} \]

  - numerator = minimum inter-cluster distance
  - denominator = maximum intra-cluster distance between tracks

- Convergence efficiency is plotted as a function of Dunn index. Shows expected structure

- CPU: 0.33 ± 0.02
- QPU: 0.56 ± 0.14
Performance on 100 events with 3 p-p collisions and 15 tracks

Performance on an ensemble of events

- 100 events with 3 p-p collisions and 15 tracks are thrown from measured CMS distributions
- Each event is sampled 10,000 times by both the QPU and the CPU (in equivalent time)
- Events with collisions spread closely compared to the spread of their tracks are hard for both QPU and CPU to solve
- A distribution of convergence efficiencies is observed
  - QPU: mean = 42%, std. dev. = 25%
  - CPU: mean = 24%, std. dev. = 11%

Is there underlying structure to these distributions?

Performance against event “clumpiness”

- A measure of event clumpiness is the Dunn index. Low Dunn index = diffuse event, high = clumpy event

\[ D = \frac{\min_{1\leq i<j\leq n} d(i,j)}{\max_{1\leq k\leq n} d'(k)} \]

numerator = minimum inter-cluster distance
denominator = maximum intra-cluster distance between tracks

- Convergence efficiency is plotted as a function of Dunn index. Shows expected structure

Convergence efficiency increases with event clumpiness. QPU beats CPU in efficiency for same running time
QPU vs CPU scaling with event complexity

We scan over event topologies with increasing complexity

- **2 collisions, 10 tracks**
  - CPU: 0.94 ± 0.01
  - QPU: 0.98 ± 0.01

- **2 collisions, 16 tracks**
  - CPU: 0.46 ± 0.01
  - QPU: 0.96 ± 0.01

- **4 collisions, 12 tracks**
  - CPU: 0.32 ± 0.02
  - QPU: 0.24 ± 0.12

- **4 collisions, 16 tracks**
  - CPU: 0.17 ± 0.01
  - QPU: 0.08 ± 0.08

Maximum convergence efficiencies

- **2 collisions, 10 tracks**
  - CPU: 0.94 ± 0.01
  - QPU: 0.98 ± 0.01
QPU vs CPU scaling with event complexity

We scan over event topologies with increasing complexity

- **2 collisions, 10 tracks**
  - QPU: $0.91 \pm 0.23$
  - CPU: $0.84 \pm 0.27$

- **2 collisions, 16 tracks**
  - QPU: $0.79 \pm 0.24$
  - CPU: $0.36 \pm 0.2$

- **4 collisions, 12 tracks**
  - QPU: $0.52 \pm 0.12$
  - CPU: $0.23 \pm 0.11$

- **4 collisions, 16 tracks**
  - QPU: $0.22 \pm 0.05$
  - CPU: $0.07 \pm 0.04$

Maximum convergence efficiencies
- **CPU:** $0.94 \pm 0.01$
- **QPU:** $0.98 \pm 0.01$
- **CPU:** $0.46 \pm 0.01$
- **QPU:** $0.96 \pm 0.01$
- **CPU:** $0.32 \pm 0.02$
- **QPU:** $0.24 \pm 0.12$
- **CPU:** $0.17 \pm 0.01$
- **QPU:** $0.08 \pm 0.08$

- QPU has advantage at low complexity. Why?
- Can any measure highlight the tunneling advantage?
QPU vs CPU scaling with event complexity

- One measure of complexity: Number of logical qubits used = number of collisions × number of tracks
- Trend: Asymptotic maximum of convergence efficiency plotted against logical qubits
  - Spread of maximum convergence efficiency represented by uncertainty bars

QPU performance comparable to a modern CPU
QPU running may be further optimized
Conclusions

- The D-Wave 2000Q_2_1 QPU can reconstruct p-p collision positions at hadron colliders in a limited capacity
- The Tevatron had ~ 3 p-p collisions per event. Would have been possible with D-Wave
- QPU implementation comparable to Simulated Annealing on MacBook CPU for equal time
- The D-Wave 2000Q_2_1 QPU can reconstruct p-p collision positions at hadron colliders in a limited capacity

Outlook

Two research directions to improve QPU implementation:

- Improve convergence efficiency:
  - Understand how distortion functions like g(x;m) work
- Use annealing offsets
- Tune annealing time, re-thermalization delay
- Try reverse annealing
- Optimize chip lengths and weights
- Fit larger problems on QPU:
  - Customized embedding
  - Solve larger problems with hierarchical clustering

Acknowledgements

Support and useful discussions with D-Wave: Joel Gottlieb, Mark Johnson, Alexander Condello
Support from the Department of Energy grant DE-SC0007884 and the Purdue Research Foundation

S. Das, A. J. Wildridge, S. B. Vaidya, A. W. Jung,
“Track clustering with a quantum annealer for primary vertex reconstruction at hadron colliders”
https://arxiv.org/abs/1903.08879