Project Documentation

Quantum Computing for Jet Clustering Optimization

Team apmcmlz2500206fj

December 2025

Contents

1 Overview

This project applies quantum computing techniques to solve the jet clustering problem in high-energy physics. We formulate the problem as a QUBO (Quadratic Unconstrained Binary Optimization) model and solve it using both classical algorithms and Kaiwu SDK quantum-inspired solvers. The solution is validated on real quantum hardware (Bose Quantum CPQC-550).

2 Data Files

2.1 Input Data

- `events_data.npz`
  Purpose: Compressed NumPy binary file containing 50 simulated particle collision events.
  Contents: Particle properties (p_T, η, ϕ) and pre-computed ΔR distance matrices.

- `events_csv/events_table.csv`
  Purpose: CSV format of particle events data (682 rows, 50 events).
  Fields: `event_id`, `particle_id`, `pT`, `eta`, `phi`.

- `events_csv/generation_params.csv`
  Purpose: Data generation parameters for reproducibility.
  Contains: Random seed (2025), particle count range (12-15), p_T range, η range.

2.2 Configuration Files

- `real_event0_lam200_mu30.json`
  `real_event0_lam200_mu100_tmax100.json`
  `real_event0_lam300_mu60_tmax100.json`
  `real_event0_lam300_mu60_tmax100_oh1.json`
  `real_event1_lam300_mu60_tmax100_oh1.json`
  `real_event2_lam300_mu60_tmax10_oh1.json`
  Purpose: JSON configuration files for QUBO model parameters and quantum solver results.
  Parameters: λ (constraint penalty), μ (balance coefficient), solution vectors.

2.3 Result Files

- `results_baseline_sa.csv`
  Purpose: Results from classical Simulated Annealing solver on all 50 events.
  Metrics: Best energy, mean energy, std energy, solve time, feasibility.

- `results_baseline_random.csv`
  Purpose: Results from random search baseline on all 50 events.

- `results_kaiwu_sa.csv`
  Purpose: Results from Kaiwu SDK Simulated Annealing solver on all 50 events.

- `results_baseline.csv`
  Purpose: Results from K-means clustering baseline method.

- `kaiwu_runs/*.pkl`
  Purpose: Pickled Python objects containing detailed Kaiwu solver outputs.
  Contains: Full solution history, energy trajectories, solver parameters.

3 Python Scripts

3.1 Data Generation

- `generate_data.py`
  Purpose: Generate synthetic particle collision events for testing.
  Outputs: `events_data.npz` and `events_csv/` directory.
  Key Functions:
  - `generate_toy_events()`: Creates random particle events
  - `deltaR_matrix()`: Computes ΔR distance matrix
  - `save_events_npz()`: Saves data in NumPy format
  - `save_events_csv()`: Saves data in CSV format

3.2 QUBO Model Construction

- `qubo_model.py`
  Purpose: Basic QUBO model construction (deprecated, use `real_qubo_model.py`).
  Function: `build_qubo_matrix_k2()`: Builds QUBO matrix for K=2 clusters.

- `real_qubo_model.py`
  Purpose: Production QUBO model with constraint handling and balance terms.
  Key Functions:
  - `build_qubo_matrix_k2()`: Constructs QUBO matrix with constraints
  - Parameters: λ (constraint), μ (balance), w (distance weighting)

3.3 Baseline Solvers

- `baseline_random.py`
  Purpose: Random search baseline solver.
  Method: Randomly samples feasible solutions (50,000 samples).
  Output: `results_baseline_random.csv`.

- `baseline_sa.py`
  Purpose: Classical Simulated Annealing solver.
  Method: Custom SA implementation with adaptive temperature schedule.
  Parameters: 10,000 steps, 10 runs per event.
  Output: `results_baseline_sa.csv`.

- `baseline_kmeans.py`
  Purpose: K-means clustering baseline (non-QUBO method).
  Method: Clusters particles in (η, ϕ) space.
  Output: `results_baseline.csv`.

3.4 Kaiwu SDK Solvers

- `baseline_kaiwu_sa.py`
  Purpose: Kaiwu SDK Simulated Annealing solver (simulation mode).
  Method: Uses Kaiwu's quantum-inspired SA solver.
  Parameters: 10,000 sweeps, 10 reads per event.
  Output: `results_kaiwu_sa.csv`.

- `kaiwu_config.py`
  Purpose: Configuration module for Kaiwu SDK authentication.
  Contains: API credentials, platform endpoints.

- `run_real_event0_cpqc550.py`
  `run_real_event1_cpqc550.py`
  `run_real_event2_cpqc550.py`
  Purpose: Scripts for submitting jobs to real quantum hardware (CPQC-550).
  Method: Connects to Bose Quantum cloud platform.
  Output: JSON files with quantum solver results.

3.5 Analysis and Visualization

- `analyze_results.py`
  Purpose: Statistical analysis of solver performance across all events.
  Generates: Summary statistics, performance comparisons.

- `compare_solvers.py`
  Purpose: Multi-dimensional comparison of different solvers.
  Metrics: Energy quality, solve time, feasibility, stability.

- `plot_energy_vs_time_sa_vs_kaiwu.py`
  Purpose: Generates energy vs. time plots comparing SA and Kaiwu solvers.
  Output: `energy_vs_time_sa_vs_kaiwu.png/pdf`.

- `plot_jet_efficiency_hist.py`
  Purpose: Plots jet clustering efficiency histograms.

- `make_paper_plots.py`
  Purpose: Master script for generating all publication-quality figures.
  Outputs: All plots in `figures/` directory (PNG and PDF formats).

4 Output Directories

4.1 Results and Figures

- `figures/` (Chinese: /)
  Purpose: Directory containing all publication figures and tables.
  Contents:
  - Energy comparison plots (histograms, boxplots, scatter)
  - Solve time histograms
  - Pareto fronts (energy vs. time)
  - Method illustration diagrams
  - Result comparison tables (CSV format)

- `kaiwu_runs/`
  Purpose: Directory storing Kaiwu solver output files (.pkl format).
  Naming: `WuYueCup_real_event{idx}_N{N}_lam{lam}_mu{mu}_..._results.pkl`.

- `screenshots/` (Chinese: /)
  Purpose: Screenshots from Bose Quantum cloud platform.
  Contents:
  - `task_list.png` (Chinese: .png): Task list screenshot
  - `usage_overview.png` (Chinese: .png): Usage overview screenshot

5 External Dependencies

5.1 Kaiwu SDK

- `kaiwu-1.3.0-cp310-none-any.whl`
  Purpose: Kaiwu SDK wheel package for Python 3.10.
  Installation: `pip install kaiwu-1.3.0-cp310-none-any.whl`.
- `kaiwu-pytorch-plugin/`
  Purpose: Kaiwu PyTorch plugin source code and examples.
  Contains: Documentation, example notebooks, test scripts.

6 Workflow Summary

6.1 Data Generation

1. Run `generate_data.py` → Creates `events_data.npz` and `events_csv/`

6.2 Baseline Solving

1. Run `baseline_random.py` → `results_baseline_random.csv`
2. Run `baseline_sa.py` → `results_baseline_sa.csv`
3. Run `baseline_kmeans.py` → `results_baseline.csv`
4. Run `baseline_kaiwu_sa.py` → `results_kaiwu_sa.csv`

6.3 Quantum Hardware Solving

1. Configure `kaiwu_config.py` with credentials
2. Run `run_real_event0_cpqc550.py` → JSON output
3. Run `run_real_event1_cpqc550.py` → JSON output
4. Run `run_real_event2_cpqc550.py` → JSON output

6.4 Analysis and Visualization

1. Run `analyze_results.py` → Statistical summaries
2. Run `compare_solvers.py` → Comparison metrics
3. Run `make_paper_plots.py` → All figures in `figures/`

7 File Statistics

8 Key Technologies

- **Programming Language:** Python 3.10+
- **Core Libraries:** NumPy, Pandas, Matplotlib
- **Optimization Methods:** QUBO, Simulated Annealing, K-means
- **Quantum Platform:** Bose Quantum CPQC-550 (550 qubits)
- **SDK:** Kaiwu SDK v1.3.0

Category	**Count**
Python scripts (.py)	13
Data files (NPZ, CSV)	3
Configuration files (JSON)	6
Result files (CSV)	4
Solver output files (PKL)	5
Figures (PNG, PDF)	31
Tables (CSV)	3

Table 1: Project file statistics