QUBO-BASED TRACK RECONSTRUCTION

JAKOB JERIČ

Fakulteta za matematiko in fiziko
Univerza v Ljubljani

This article considers the reconstruction of particle tracks in high-energy detectors as a global optimisation problem. Detector hits are combined into local geometric segments and represented by binary variables within a quadratic unconstrained binary optimisation (QUBO) model. The QUBO coefficients encode geometric consistency and mutual compatibility between neighboring segments. The resulting optimisation problem is solved using simulated annealing, and the reconstruction performance is evaluated in terms of purity and efficiency. The approach achieves high reconstruction quality, while the dominant computational cost arises from the construction of the QUBO model itself. The article concludes by outlining the potential for further optimisation and the prospect of more favorable scaling using quantum optimisation methods.

REKONSTRUKCIJA DELČNIH SLEDI Z BINARNO OPTIMIZACIJO (QUBO)

Članek obravnava rekonstrukcijo delčnih sledi v detektorjih visokih energij kot globalni optimizacijski problem. Detektorski zadetki so združeni v lokalne geometrijske segmente, ki so predstavljeni z binarnimi spremenljivkami v okviru kvadratne binarne neomejene optimizacije (QUBO). Koeficienti QUBO modela opisujejo geometrijsko skladnost in medsebojno združljivost sosednjih segmentov. Nastali optimizacijski problem se rešuje s simuliranim ohlajanjem, rezultati pa so ovrednoteni z merama čistosti in učinkovitosti rekonstrukcije. Pristop dosega visoko kakovost rekonstruiranih sledi, vendar glavna računska omejitev ostaja konstrukcija samega QUBO modela. V zaključku je poudarjen potencial nadaljnjih optimizacij ter možnost ugodnejšega skaliranja z uporabo kvantnih optimizacijskih metod.

1. Introduction

At the Large Hadron Collider (LHC) [1], protons are accelerated to nearly the speed of light and collided at centre-of-mass energies of $\sqrt{s} \approx 13$ TeV (Run 3 conditions) at defined interaction points inside massive detectors such as ATLAS [2] and CMS [3]. Each collision produces hundreds or thousands of charged particles that traverse concentric layers of silicon detectors immersed in a strong magnetic field. As a charged particle passes through a silicon layer, it deposits charge that appears as a localized ionization signal, referred to as a *hit* (Figure 1 (a)). Each hit thus represents a spatial measurement of where a particle interacted with the detector material.

The goal of track reconstruction is to group these hits into coherent trajectories, or *tracks* (Figure 1 (b)), that best describe the paths of the original particles. Mathematically, this represents a massive combinatorial pattern recognition problem: isolating the true, continuous physical trajectories from an overwhelmingly large point cloud of noise and overlapping signals. From these tracks, physicists infer particle properties such as momentum, charge sign, and production vertex. Accurate track reconstruction is indispensable for nearly all physics analyses at the LHC, forming the basis for particle identification, vertexing, and precision measurements [4].

1.1 Current reconstruction methods

Modern track reconstruction in collider experiments is dominated by algorithms based on the Kalman filter [5], a recursive method that combines prediction and measurement to estimate a particle's trajectory. The procedure begins with seeding, where a small number of hits define an initial estimate of the direction and curvature in the magnetic field. This state vector is then propagated through successive detector layers, predicting where the next hit should occur. When a

compatible measurement is found, the filter updates the state by weighting prediction and measurement according to their uncertainties. In this way, trajectory parameters—position, direction, and momentum—are refined progressively as the particle crosses additional layers [5, 4].

To handle ambiguous cases where several hits match a track hypothesis, multiple candidate trajectories are propagated in parallel; this strategy underlies the reconstruction chains used in CMS and ATLAS [6]. After all hits are processed, a smoothing step runs the filter forward and backward along each candidate, yielding the most precise estimate of the kinematic parameters. The Kalman filter remains the workhorse of modern tracking due to its ability to naturally incorporate multiple scattering and energy loss while maintaining stability in the presence of detector noise [4].

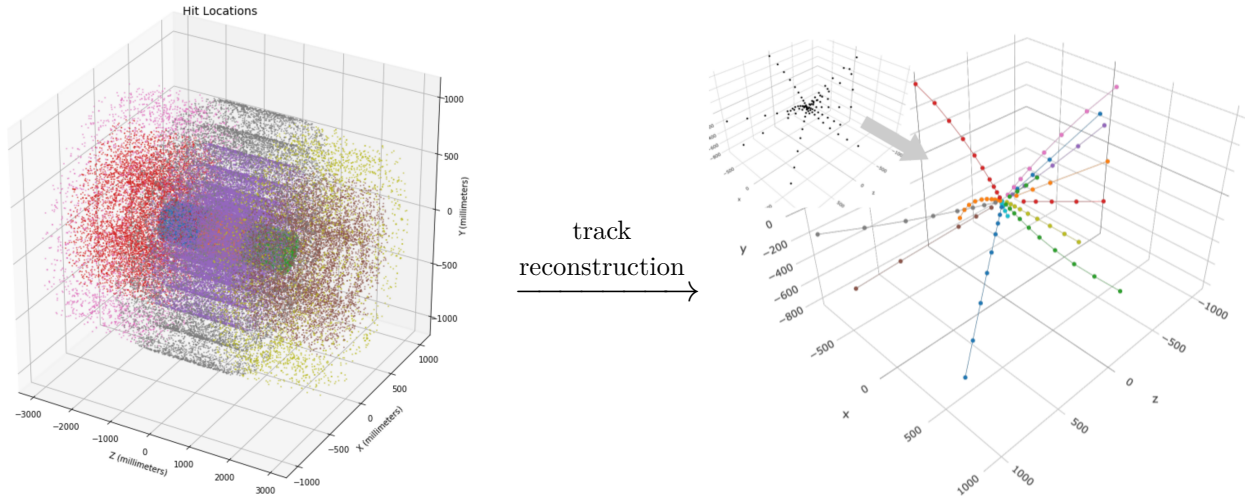


Figure 1. From detector hits from a single TrackML event (left) to reconstructed tracks from a subset of hits (right) via track finding [7].

1.2 Computational challenges and future outlook

As detector granularity and collision rates increase, track reconstruction has become one of the most computationally demanding tasks in high-energy physics. During Run 2, ATLAS reported that tracking accounted for about 60–65% of offline reconstruction CPU time, and despite extensive optimisation it still represents roughly 40% in early Run 3 [8].

In current Run 3 operations, the average number of inelastic proton–proton interactions per bunch crossing, $\langle\mu\rangle$, typically reaches around 60 and can exceed this under high-luminosity conditions [9]. This simultaneous occurrence of multiple proton–proton collisions in a single event is referred to as *pile-up* (illustrated in Figure 2). As pile-up ($\langle\mu\rangle$) rises, the number of possible hit-to-track associations grows combinatorially, and reconstruction must still account for multiple scattering, energy loss, and interactions with detector material. These factors together make tracking one of the most computationally intensive components of LHC data processing [4].

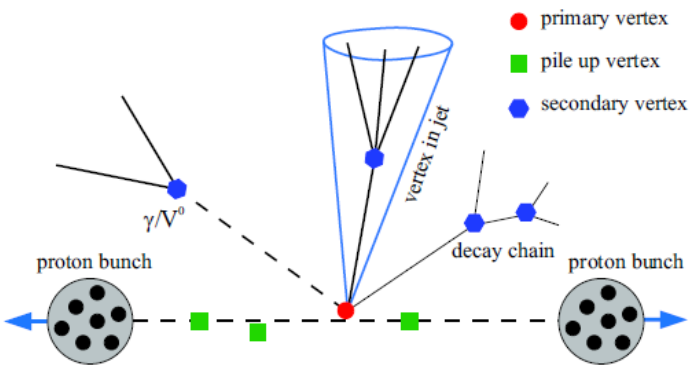


Figure 2. Pile-up illustration [7].

Looking ahead to the High-Luminosity LHC (HL-LHC), an upgrade scheduled to begin opera-

tions in June 2030 [10], pile-up levels of $\langle\mu\rangle \approx 140$–$200$ are expected [11]. Such extreme occupancies will greatly increase the number of hits per event and the combinatorial complexity of candidate trajectories, placing heavy demands on current computing architectures. Looking even further ahead, proposed next-generation facilities like the Future Circular Collider (FCC-hh) [12] will operate at significantly higher energies and luminosities, creating tracking environments of unprecedented density.

To sustain reconstruction performance across these imminent and long-term horizons, ongoing research focuses on exploring new computational paradigms. Alongside GPU-accelerated Kalman filters [13] and machine-learning methods, alternative formulations are being developed that cast tracking as a *Quadratic Unconstrained Binary Optimisation* (QUBO) problem [14]. While this QUBO formulation can be evaluated on classical hardware today, it is intrinsically designed to be solved on quantum annealers. The extended developmental timeline of future colliders like the FCC-hh therefore provides a timescale over which these quantum algorithms and hardware may mature into practical reconstruction tools.

1.3 QUBO-based track reconstruction

At a basic level, a QUBO model provides a simple way to describe a decision problem using binary variables and a cost function that rewards consistent choices and penalises incompatible ones [14]. To apply this to track reconstruction, the problem must be translated into geometric relations between individual hits. Detector hits are first combined into short, localized track segments: pairs of hits are linked into *doublets*, and pairs of compatible doublets are merged into *triplets*. These triplets are then translated into the binary variables of the QUBO model, and their geometric consistency is encoded into the coefficients of the cost function.

The specific QUBO construction used here follows the approach described and implemented in the master's thesis of Linder [7]. The objective of this article is twofold: to motivate the QUBO-based approach as a potential solution to future tracking challenges, and to verify its practical performance. The theoretical framework, geometric selections, and the foundational software implementation are adopted directly from the original work by Linder [7]. By applying this established framework to the TrackML dataset, this study reproduces the original findings, assesses the reconstruction quality, and examines the computational scaling. The adapted codebase utilised for this evaluation is available in a public GitHub repository [15].

2. The QUBO algorithm

The Quadratic Unconstrained Binary Optimisation (QUBO) framework provides a compact and flexible way to express combinatorial optimisation problems using binary variables. In the context of track reconstruction, each variable represents a potential local track element, and the goal is to select the subset of these elements that forms the most physically consistent trajectories. Instead of building tracks sequentially, the QUBO formulation evaluates many possible combinations at once and identifies the configuration that minimises a global cost function.

Mathematically, a QUBO is written as

$$E(\mathbf{a}, \mathbf{b}, \mathbf{T}) = \sum_i a_i T_i + \sum_{i<j} b_{ij} T_i T_j,$$

where $T_i \in \{0, 1\}$ are binary variables, a_i are *bias weights* favouring or suppressing individual candidates, and b_{ij} are *coupling strengths* encoding pairwise compatibility. Here each variable corresponds to a local track segment constructed from a triplet of hits, and the energy $E(\mathbf{T})$ measures how well a chosen subset satisfies the relevant geometric and topological constraints.

Optimisation consists of finding the binary vector

$$\mathbf{T}^* = \arg \min_{\mathbf{T} \in \{0,1\}^N} E(\mathbf{T}),$$

which represents the most coherent global assignment of triplets. Variables with $T_i = 1$ are included in the reconstruction, and those with $T_i = 0$ are excluded. The minimum-energy configuration therefore corresponds to the set of triplets most compatible with forming real particle tracks through the detector.

Figure 3 summarises the reconstruction pipeline, from the preparation of detector hits to the construction and solution of the QUBO model.

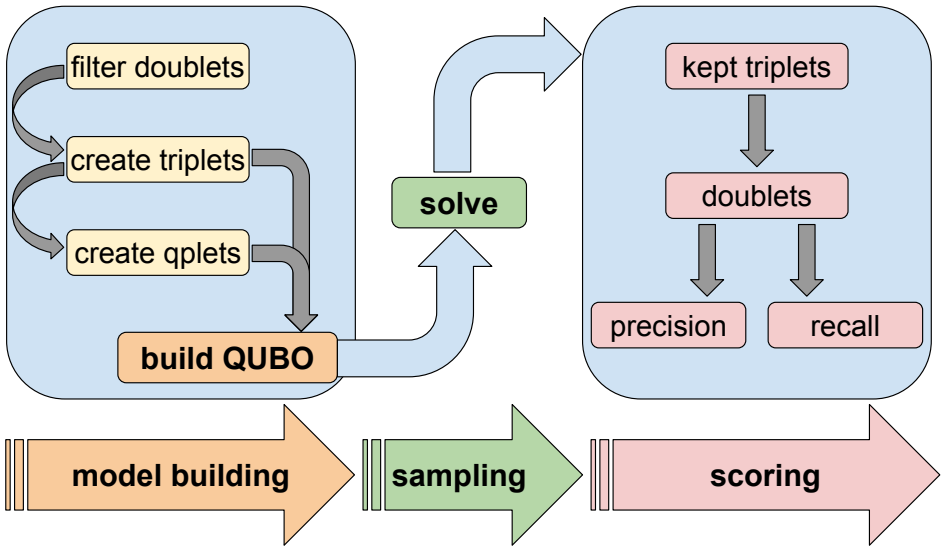


Figure 3. Overview of the QUBO-based track-reconstruction pipeline. Recreated from [7].

2.1 QUBO building

2.1.1 Dataset

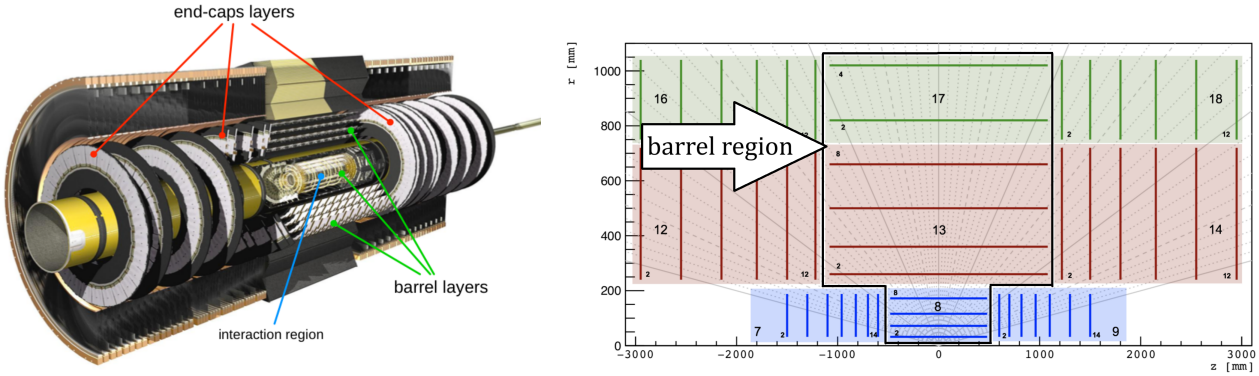
To formulate the QUBO-based reconstruction, we begin with a dataset of detector *hits*, three-dimensional space points recording where charged particles interacted with the silicon layers. These hits form the raw inputs from which doublets, triplets, and ultimately the QUBO model are constructed.

The dataset used for this reconstruction is based on the *TrackML Particle Tracking Challenge* [16], a benchmark widely employed in reconstruction studies and also used in the master's thesis of Linder [7]. It closely reflects the conditions expected at the High-Luminosity LHC, with pile-up levels of $\langle \mu \rangle = 140$–$200$ and a simplified detector geometry (Figure 4b). Each simulated event contains on the order of 10^4 charged particles and up to 10^5 silicon hits, providing a dense and realistic environment for evaluating tracking algorithms.

The reconstruction focuses on tracks with transverse momentum $p_\mathrm{T} > 1$ GeV, which bend only mildly in the magnetic field and therefore produce nearly straight local segments. Restricting to high-p_T tracks reduces geometric ambiguity while preserving the physically most relevant trajectories. In addition, only tracks with at least five hits are considered, ensuring that the reconstructed segments correspond to sufficiently long and well-defined trajectories.

To further limit complexity, only hits from the tracker *barrel region* (Figure 4a) are considered. The barrel consists of concentric cylindrical layers around the beam axis, providing a clean and regular geometry in which the radial layer index increases monotonically. This structure simplifies

the definition of spatial relationships between hits and supports efficient construction of doublets and triplets, while still capturing the main challenges posed by HL-LHC level occupancies.



(a) **(b)**

Figure 4. Simplified tracker geometry illustrating the cylindrical barrel region: generic schematic (a) ([7], adapted from [17]) and the TrackML detector layout with marked barrel region (b), adapted from [18].

2.1.2 Doublets, triplets, and quadruplets

After preprocessing the detector hits, the next step is to build small geometric objects that describe how a particle may have moved through successive layers. The simplest are *doublets*, constructed by connecting two hits on adjacent or nearly adjacent layers, ordered by increasing radius from the beam axis. Each doublet represents a short local segment of a possible trajectory. To restrict the large number of potential connections, basic geometric cuts are applied: the azimuthal and polar angle differences $\Delta\phi$ and $\Delta\theta$ must lie within reasonable bounds, and the radial separation must be consistent with the expected motion of a charged particle. Connections that skip at most one intermediate layer are permitted to account for occasional missing hits. These filters suppress random combinations while retaining physically plausible ones.

From the accepted doublets, *triplets* $T_{abc} = (a, b, c)$ are formed by joining two compatible doublets (a, b) and (b, c) that share the middle hit. Each triplet is then checked for smoothness and continuity: the two doublets must align so that the trajectory changes only gradually, with small angular deviation in the RZ-plane and a curvature consistent with the bending expected from a high-p_{T} (> 1 GeV) particle. Figure 6 illustrates these geometric features, showing how high-p_{T} and low-p_{T} tracks differ both in the transverse plane and in the RZ-projection. Triplets that exhibit sharp kinks or incompatible curvature are discarded, leaving a compact set of geometrically coherent segments.

Compatible triplets can be merged into *quadruplets* (or *q-plets*) when they share two consecutive hits and extend the same smooth trajectory. These structures strengthen the physical consistency of the reconstruction by linking several layers at once. The quality of each x-plet, doublet, triplet, or quadruplet, is characterised by its curvature, its local slope in the RZ-plane, and the number of missing layers (holes). These act as *quality cuts*: tighter cuts yield a smaller, cleaner set of nearly straight high-p_{T} candidates, while looser ones admit more strongly bent, lower-p_{T} tracks at the cost of increased combinatorics. The resulting validated triplets and quadruplets, illustrated in Fig. 5, provide a noise-reduced representation of possible track segments and form the basis for the subsequent QUBO modelling stage.

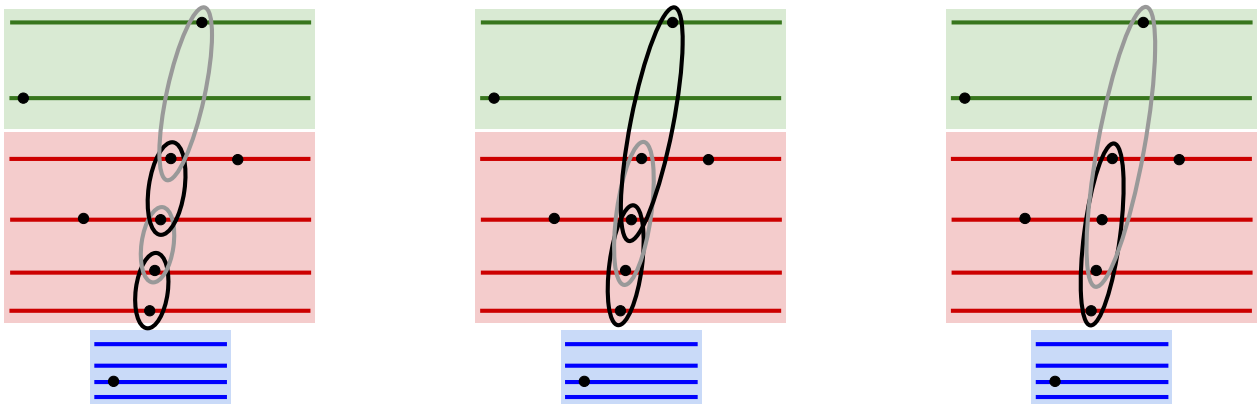


Figure 5. Construction of doublets (a), triplets (b) and quadruplets (c) from hits on successive barrel layers of the TrackML detector.

2.1.3 QUBO formulation

Once the geometrically valid triplets and quadruplets have been constructed, the next step is to translate them into the coefficients of the QUBO model. Each binary variable T_i corresponds to a triplet T_{abc}, and the goal is to favour triplets that follow smooth high-p_{T} trajectories while suppressing combinations that cannot belong to the same physical track. The linear terms a_i and quadratic couplings b_{ij} are therefore derived directly from the geometric relations between triplets. The same quantities that define the x-plet quality, transverse curvature, the local RZ-slope, and the number of missing layers (holes), enter the QUBO coefficients.

The quadratic terms b_{ij} encode whether two triplets can serve as a coherent continuation of one another. If two triplets share their last two hits, such as (a, b, c) and (b, c, d), they define a quadruplet spanning four layers. The quality of this extension is captured by the quadruplet strength

$$S(T_i, T_j) = \frac{1 - \frac{1}{2}\left(|\delta(\mathrm{curv}_i, \mathrm{curv}_j)| + \max(\mathrm{drz}_i, \mathrm{drz}_j)\right)}{(1 + H_i + H_j)^2}.$$

Here $\mathrm{curv}_i = 1/R_i$ is the curvature of triplet T_i, obtained from the radius R_i of the circle fitted through its three hits in the transverse xy-plane (Fig. 6a and 6b), and $\delta(\mathrm{curv}_i, \mathrm{curv}_j)$ measures how different the two curvatures are. The quantity drz characterises the kink of a triplet in the RZ-projection (Fig. 6c); for $T_{a,b,c} = T_i$ we define $drz_i = \left| \frac{\Delta z_{ab}}{\Delta r_{ab}} - \frac{\Delta z_{bc}}{\Delta r_{bc}} \right|$, where the doublets are (a, b) and (b, c). Small curvature differences, small longitudinal kink, and few holes (H_i) lead to large $S(T_i, T_j)$, indicating geometrically consistent quadruplets, while mismatches produce a small strength and are disfavoured in the QUBO.

Using this strength, the QUBO couplings are defined as

$$b_{ij} = \begin{cases} -S(T_i, T_j), & \text{if } (T_i, T_j) \text{ form a valid quadruplet,} \\ \zeta, & \text{if the triplets are in conflict,} \\ 0, & \text{otherwise,} \end{cases}$$

where ζ is a fixed hyperparameter enforcing the *no shared hits* constraint. A conflict occurs when two triplets share a hit in a way inconsistent with a single non-branching track. Thus, the quadratic terms reward smooth, physically plausible extensions and penalise incompatible overlaps.

The linear terms a_i set the intrinsic preference for selecting individual triplets. In the simplest variant all valid triplets are weighted equally,

$$a_i = a \qquad \text{for all } i,$$

so the structure of the solution is driven mainly by pairwise compatibilities. With this choice, the QUBO energy becomes

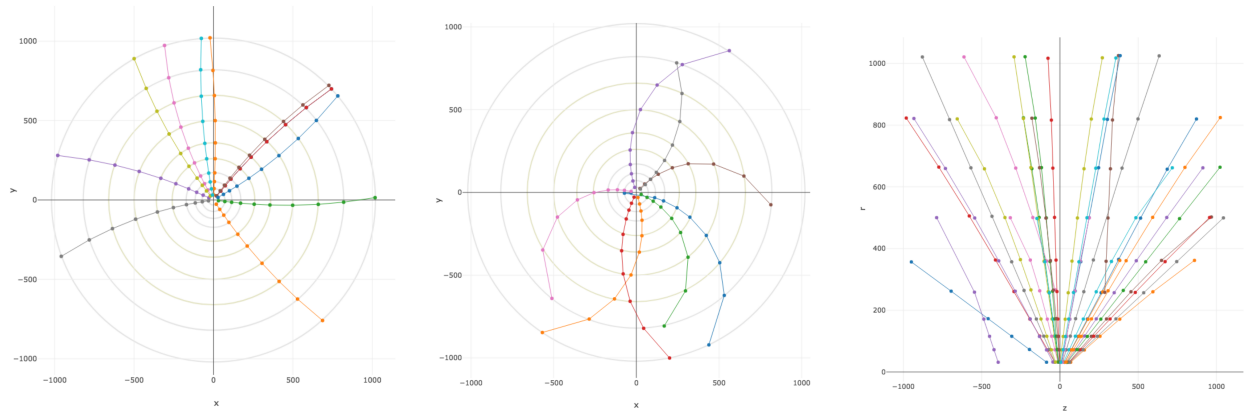
$$E(\mathbf{a}, \mathbf{b}, \mathbf{T}) = \sum_i a\, T_i + \sum_{i<j} b_{ij}\, T_i T_j, \tag{1}$$

so that the global minimum corresponds to the most coherent collection of triplets.

A more refined formulation introduces a bias based on how well each triplet points back to the primary interaction point. The transverse impact parameter d_0 is defined as the distance of closest approach of the fitted trajectory to the beamline in the transverse plane, while the longitudinal impact parameter z_0 is the z-coordinate of the trajectory at this point of closest approach, evaluated by extrapolating the constituent doublets in the RZ-projection. These are combined into the smooth penalty function

$$a_i = \alpha \left(1 - e^{-|d_0|/\gamma}\right) + \beta \left(1 - e^{-|z_0|/\lambda}\right),$$

where the parameters α, β, γ, and λ set the strength and characteristic scale of the penalties. Triplets compatible with originating near the interaction region receive small biases, while misaligned or fake triplets are increasingly suppressed.



(a) **(b)** **(c)**

Figure 6. Examples of high-p_T (a) and low-p_T (b) trajectories in the transverse (x, y) plane and their projection in the RZ-plane (c) [7].

2.2 QUBO solving

With the QUBO coefficients defined, the task is to find the binary vector $\mathbf{T}$ that minimises the energy in Eq. (1). The ground state of this energy function corresponds to the most coherent and geometrically consistent set of triplets and thus to the most probable collection of particle tracks. Because the search space grows exponentially with the number of variables, a range of classical and quantum optimisation techniques can be employed to approximate or approach this minimum.

2.2.1 Classical simulated annealing

There exist many approximate optimisation methods for minimising high-dimensional cost functions, and one of the most widely used in the context of QUBO problems is *simulated annealing*, a stochastic technique that mimics the thermal annealing of physical systems [19, 20]. Starting from an initial configuration of the binary variables, the algorithm proposes random updates and evaluates the corresponding change in energy ΔE. Moves that reduce the energy are always accepted, while uphill moves are accepted with probability $\exp(-\Delta E/T)$, where T is an effective temperature parameter. At high temperature, the acceptance probability is large, allowing the algorithm to

explore the energy landscape broadly and escape shallow local minima. As the temperature is gradually lowered according to a predefined schedule, uphill moves become increasingly unlikely, and the system settles into a low-energy configuration. Applied to Eq. (1), simulated annealing provides a flexible and hardware-independent way of searching the exponentially large configuration space for a near-optimal set of triplets.

2.2.2 Quantum annealing

Quantum annealing aims to find a low-energy configuration of the QUBO by exploiting the adiabatic evolution of a quantum system. The method introduces two Hamiltonians: an initial Hamiltonian H_0, whose ground state is easy to prepare and corresponds to a uniform superposition over all bitstrings, and the problem Hamiltonian H_P, whose ground state encodes the minimum of Eq. (1). The system is evolved under a time-dependent Hamiltonian

$$H(s) = (1 - s)\, H_0 + s\, H_P,$$

where $s = t/t_f \in [0, 1]$ parametrizes the annealing time. At $s = 0$ the Hamiltonian is entirely H_0, while at $s = 1$ it becomes H_P. In the ideal limit of an isolated system and infinitely slow interpolation, and provided the instantaneous energy gap between the ground and excited states does not close, the adiabatic theorem states that a system prepared in the ground state of H_0 will remain in the instantaneous ground state of $H(s)$ throughout the evolution. In practice, the anneal time is finite and the device is subject to noise and decoherence, so the state only approximately follows the ground state and ends up dominated by low-lying energy eigenstates rather than the exact ground state.

At the beginning of the anneal, the system occupies the ground state of H_0. As s increases, the contribution of H_P gradually shapes the energy landscape, guiding the system toward low-energy configurations. The difference between the classical and quantum annealing schemes is illustrated in Fig. 7. At the end of the anneal ($s = 1$), the qubits are measured, yielding a classical bitstring that approximates the ground state of H_P.

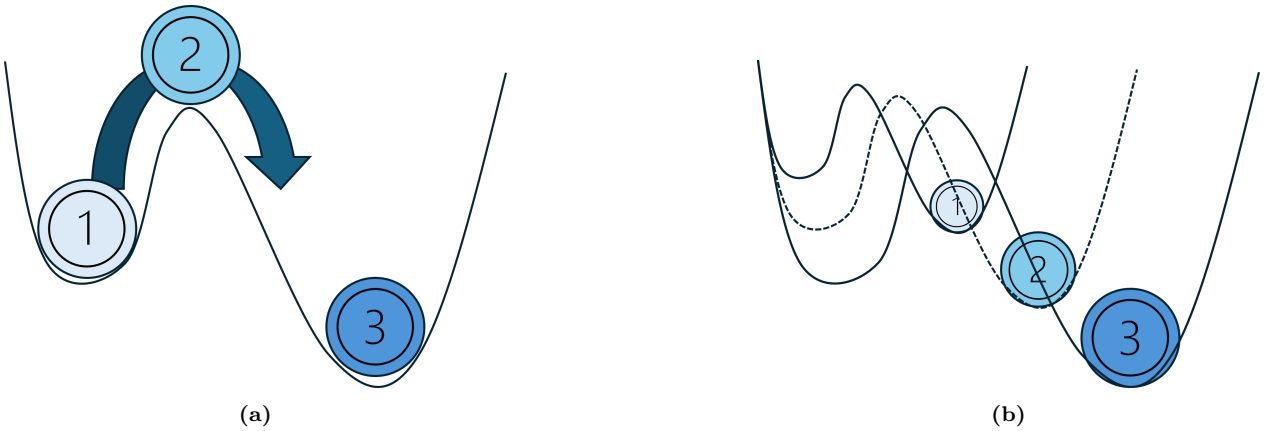


Figure 7. Conceptual illustration of simulated annealing (a) and quantum annealing (b). Recreated from [21].

2.3 Results

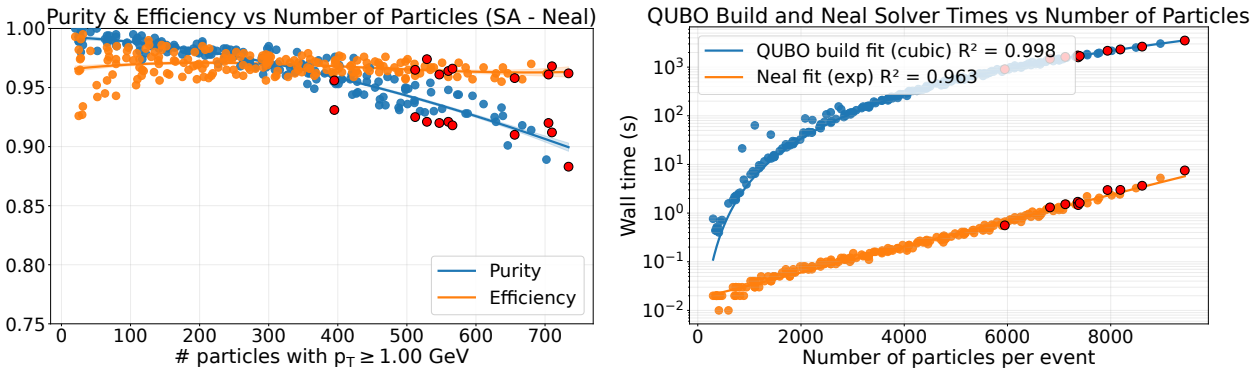
After solving the QUBO, the binary vector $\mathbf{T}$ indicates which triplets T_{abc} appear in the minimum-energy configuration. These triplets are linked into *track candidates* by connecting triplets that share two consecutive hits, following the triplet–quadruplet logic described earlier. Each candidate is then decomposed into its constituent doublets, and only doublets belonging to candidates with at least five hits are kept for evaluation.

The selected doublets are compared against the TrackML ground truth (Section 2.1.1). A reconstructed doublet is counted as true if both hits originate from the same simulated particle; otherwise it is classified as fake. Reconstruction quality is quantified using

$$\text{Purity} = \frac{N_{\text{true}}}{N_{\text{true}} + N_{\text{fake}}}, \qquad \text{Efficiency} = \frac{N_{\text{true}}}{N_{\text{true}}^{(\text{truth})}},$$

where $N_{\text{true}}^{(\text{truth})}$ denotes the number of ground-truth doublets that satisfy the same geometric selections. Purity measures the fraction of reconstructed segments that correspond to real particle trajectories, while efficiency expresses how many physically valid doublets are recovered.

The results shown in Figure 8 were obtained using the Python implementation of the full QUBO-based reconstruction pipeline available in the public GitHub repository [15]. The code closely follows the original implementation by Linder [7]. For each event, the pipeline constructs doublets and triplets from the TrackML hits, builds the corresponding QUBO, solves it using the simulated-annealing solver `Neal`, and evaluates the resulting segments against the ground truth. The plots therefore reflect fully end-to-end performance, from raw hits to reconstructed doublets and their purity and efficiency. In the figures, red markers denote full events (ds = 100), while the remaining points correspond to reduced-density versions of the same events, in which only a subset of the hits is used for reconstruction.



(a) **(b)**

Figure 8. Reconstruction quality (a) and wall-time scaling (b) for the QUBO-based reconstruction using the simulated-annealing solver `Neal`. Red points denote full events (ds = 100).

The quality plot shows that efficiency stays above 0.95 for all evaluated events and purity exceeds 0.90 in nine out of ten cases, demonstrating that the QUBO formulation yields stable reconstruction performance across events of varying occupancy.

The timing plot highlights the main motivation for exploring quantum annealing: classical simulated annealing exhibits exponential growth with problem size, while the benchmarking study of Kim et al. [22] reports that modern quantum annealers maintain nearly constant anneal time even for dense QUBOs with up to 10^4 variables. Their hybrid approach also outperforms classical solvers by large factors (up to 6561-times), suggesting that, with sufficiently large embeddings, QUBO-based tracking could benefit from substantially more favourable scaling.

Currently, however, the dominant cost lies in *constructing* the QUBO. Forming doublets, triplets, and assembling the full coefficient matrix, following the implementation of [7], scales poorly and is performed serially. As noted by Okawa et al. [14], this modelling stage is far from optimised and is expected to benefit significantly from parallelisation and algorithmic improvements, indicating that QUBO construction itself does not represent a fundamental limitation of the approach.

3. Conclusions

The QUBO-based formulation examined here offers a fundamentally different perspective on track reconstruction compared with the well-established Kalman-filter pipelines used at the LHC. Traditional methods are highly refined, robust against detector effects, and efficiently parallelised on modern multi-core architectures [13, 6]. Their scaling is well understood, though still challenged by the extreme occupancies foreseen at the HL-LHC [11]. The QUBO approach instead frames pattern recognition as a global optimisation problem, providing a conceptually distinct way of formulating reconstruction.

The results show that the QUBO model achieves high purity and recall even when solved with a classical simulated-annealing algorithm, indicating that it captures the essential geometric structure of high-p_{T} tracks. At present, however, the dominant cost lies in constructing the QUBO itself. As highlighted in [14] and confirmed by the baseline results in this study, generating doublets, triplets, and QUBO coefficients is traditionally carried out serially and remains far from optimised. However, ongoing developments in the associated software repository [15] have demonstrated that parallelising these data-preparation stages reduces the QUBO construction time by approximately an order of magnitude. This confirms that while substantial algorithmic refinement is still required, the construction bottleneck is an implementation issue rather than a fundamental limitation of the approach.

Despite these limitations, the long-term outlook remains promising. Classical simulated annealing scales poorly, whereas modern quantum annealers exhibit nearly problem-size-independent anneal times for large QUBOs, as shown by Kim et al. [22]. If future devices can embed QUBOs of realistic detector size, and if QUBO construction can be accelerated accordingly, the approach may eventually display scaling behaviour quite different from conventional algorithms. From this perspective, the formulation is not intended as an immediate replacement for current methods, but as a promising research direction for future hybrid classical–quantum reconstruction strategies.

REFERENCES

[1] CERN, *The Large Hadron Collider*, https://home.cern/science/accelerators/large-hadron-collider (accessed 20 Nov 2025).

[2] ATLAS Collaboration, *The ATLAS Experiment*, https://atlas.cern (accessed 21 Nov 2025).

[3] CMS Collaboration, *The CMS Experiment*, https://cms.cern (accessed 22 Nov 2025).

[4] A. Strandlie and R. Frühwirth, *Track and vertex reconstruction: From classical to adaptive methods*, Reviews of Modern Physics **82** (2010), 1419–1458. https://doi.org/10.1103/RevModPhys.82.1419

[5] R. Frühwirth, *Application of Kalman filtering to track and vertex fitting*, Nuclear Instruments and Methods in Physics Research Section A **262** (1987), 444–450. https://doi.org/10.1016/0168-9002(87)90887-4

[6] W. Adam, B. Mangano, T. Speer and T. Todorov, *Track Reconstruction in the CMS Tracker*, CERN Technical Report, 2006. https://cds.cern.ch/record/934067 (accessed 17 Nov 2025).

[7] L. Linder, *Using a Quantum Annealer for particle tracking at LHC*, Master's thesis, Université de Genève, 2019. https://github.com/derlin/hepqpr-qallse (accessed 18 Nov 2025).

[8] Z. M. Schillaci, *Improvements to ATLAS Inner Detector Track Reconstruction for Run 3*, EPJ Web of Conferences **251** (2021), 03048. https://doi.org/10.1051/epjconf/202125103048

[9] C. Bernius, *ATLAS Operations: Progress and Highlights from 2023/2024*, https://ep-news.web.cern.ch/content/atlas-operations-progress-and-highlights-20232024 (accessed 23 Nov 2025).

[10] CERN, *New schedule for CERN's accelerators*, https://home.cern/news/news/accelerators/new-schedule-cerns-accelerators (accessed 19 Nov 2025).

[11] ATLAS Collaboration, *ATLAS HL-LHC Computing Conceptual Design Report*, CERN, 2020. https://cds.cern.ch/record/2729668 (accessed 25 Nov 2025).

[12] Frank Zimmerman, *Energy, Luminosity, Operation scenarios*, https://indico.cern.ch/event/1439072/contributions/6106995/attachments/2917946/5125895/FCC-hh-scenarios-2024kickoff.pdf (accessed 15 Mar 2026).

[13] G. Cerati et al., *Parallelized Kalman-Filter-Based Reconstruction of Particle Tracks on Many-Core Architectures with the CMS Detector*, arXiv:1906.02253 (2019). https://arxiv.org/abs/1906.02253

[14] H. Okawa, Q.-G. Zeng, X.-Z. Tao and M.-H. Yung, *Quantum-Annealing-Inspired Algorithms for Track Reconstruction at High-Energy Colliders*, Computing and Software for Big Science **8** (2024). https://doi.org/10.1007/s41781-024-00126-z

[15] J. Jerič, *QUBO-based Track Reconstruction*, https://github.com/JakobJeric1/QUBO-based_track_reconstruction (accessed 24 Nov 2025).

[16] TrackML Collaboration, *TrackML Particle Tracking Challenge*, https://www.kaggle.com/competitions/trackml-particle-identification (accessed 15 Nov 2025).

[17] CERN, *The ATLAS experiment*, https://vincent-hedberg.web.cern.ch/atlas/atlas.html (accessed 20 Nov 2025).

[18] M. Kiehn et al., *The TrackML high-energy physics tracking challenge on Kaggle*, EPJ Web of Conferences **214** (2019), 06037. https://doi.org/10.1051/epjconf/201921406037

[19] GeeksforGeeks, *What is Simulated Annealing?*, https://www.geeksforgeeks.org/artificial-intelligence/what-is-simulated-annealing/ (accessed 16 Nov 2025).

[20] Baeldung on Computer Science, *Simulated Annealing Explained*, https://www.baeldung.com/cs/simulated-annealing (accessed 27 Nov 2025).

[21] Quantum WA, *Quantum Annealing*, https://medium.com/@quantum_wa/quantum-annealing-cdb129e96601 (accessed 26 Nov 2025).

[22] S. Kim, S.-W. Ahn, I.-S. Suh, A. W. Dowling, E. Lee and T. Luo, *Quantum annealing for combinatorial optimization: a benchmarking study*, npj Quantum Information **11** (2025). https://doi.org/10.1038/s41534-025-01020-1