Applied Quantum Annealing for Particle Tracking: Optimisation for the HL-LHC

by

Parker S Reid

B.Sc., Saint Mary's University, 2017

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Physics
Faculty of Science

SIMON FRASER UNIVERSITY
Summer 2021

Declaration of Committee

Name: **Parker S Reid**

Degree: **Master of Science (Physics)**

Title: **Applied Quantum Annealing for Particle Tracking: Optimisation for the HL-LHC**

Committee:

Chair: Malcolm Kennett
Associate Professor, Physics

Dugan O'Neil
Supervisor
Professor, Physics

Bernd Stelzer
Committee Member
Professor, Physics

Paul Haljan
Examiner
Associate Professor, Physics

Abstract

Advancement in particle physics tracking techniques is a seemingly inevitable requirement for the future of higher luminosity experiments at the Large Hadron Collider (LHC). With the advancements in quantum annealing, it is now possible to place a minimisation based track reconstruction algorithm on a quantum computer in the form of a quadratic unconstrained binary optimisation problem (QUBO). The quantum annealing approach requires sufficient resources to generate a QUBO. Unfortunately, this QUBO is too large for current annealing hardware and must be partitioned by slicing the dataset. This has a detrimental impact on scoring metrics such as efficiency and purity, but reduces the overall runtime of the algorithm by a factor of two from the non-sliced counterpart. The ATLAS experiment is one of the experiments at the LHC. ATLAS is able to provide a simulated dataset, which can then be used to determine the effectiveness of the QUBO in a fully realistic event similar to the incoming High Luminosity Large Hadron Collider. Depending on the hard cuts applied to pre-QUBO generation for dense events, the realistic dataset leads to either a considerable drop in performance metrics, or an exponential growth in size of the QUBO. For these reasons it is probable that quantum annealing techniques in track reconstruction will remain limited until the size of quantum annealing chips (and therefore the size of the QUBO) increases.

Keywords: Particle Physics; Particle Tracking; High-Energy Physics; Quantum Annealing

Acknowledgements

I would like to thank first and foremost my supervisor Dr. Dugan O'Neil, and Dr. Eric Drechsler for the continued guidance and support throughout this process. I would also like to thank Simon Fraser University physics department, who have been invaluable in providing educational and social support throughout my tenure as a masters student. This thesis would not be possible without the hard work from Lucy Linder and her remarkable thesis and coding ability. This laid the foundation for the works presented here. I would like to thank DWave for making all of their quantum resources available during this process. Finally I would like to thank my friends and family who have made my life away from home comfortable... even during the ongoing pandemic.

Table of Contents

List of Figures

Chapter 1

Introduction

Particle physics has made substantial progress in expanding our understanding of the universe. The idea behind experimental particle physics at the Large Hadron Collder (LHC), located at the European organization for Nuclear Research (CERN) is simple: smash protons together at high energy and observe what happens. What happens in these collisions is then compared with the theory. This is achieved by using large accelerator facilities and proton collisions. Physics groups working at CERN reconstruct the particle collisions by using various detectors. With the help of reconstruction algorithms, physicists are able to identify patterns and determine the path that a particle has taken after a collision. These particle "track" reconstruction algorithms themselves are straightforward, but become computationally intensive when a greater number of particles traverse the detector at any given time. The future of CERN is to collide protons at a rate four times more than the previous record at the LHC [1]. This in turn will greatly tax the computational resources required to run a tracking algorithm. This thesis will explore a new type of quantum track reconstruction algorithm. The techniques applied in this algorithm will make use of the new quantum annealing resources that have been made available by D-Wave. It is the hope of this thesis to further shed light on the long term viability of quantum annealing applied to track reconstruction.

1.1 The Large Hadron Collider

The Large Hadron Collider (LHC) is a particle collider located at the CERN accelerator complex in Geneva, Switzerland. This 27 km underground tunnel accelerates protons in both directions with an energy of up to 6.5 TeV each. This gives a maximum possible collision energy of 13 TeV. The LHC is responsible for many discoveries in the field of particle physics, including the joint discovery of the Higgs Boson in 2012 [2] by the ATLAS and CMS collaborations.

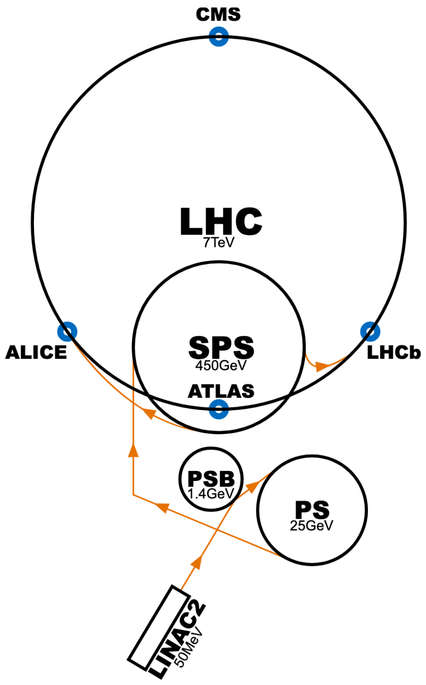


Figure 1.1: Schematic of the staging process of proton acceleration for all experiments using the LHC (not to scale), each successive synchrotron with a larger radius. [3]

1.1.1 Achieving 13 TeV Collisions in the LHC

Adding 6.5 TeV of energy to a proton is non-trivial. The task of the LHC is to energize bunches of 10^{11} protons to 6.5 TeV, and maintain them in a concentrated beam. This process requires delicate calibration of magnetic fields and uses a significant amount of energy. First, free protons are obtained by running a current through hydrogen gas, thus stripping the electrons from the system. Free protons are then bunched together and put through LINAC2, a linear accelerator that makes use of radio frequency (RF) peaks to accelerate protons to 50 MeV. These protons are then injected into the Proton Synchrotron Booster (PSB). The PSB is a circular device that allows protons to accelerate through it many times. The energy the PSB can achieve is 1.4 GeV, which is limited by the radius of the synchrotron, and the strength of the magnetic dipoles that keep the protons in a circular trajectory. Each successive stage is just a synchrotron with a larger radius. The same RF method is used to elevate the proton energy to 25 GeV in the Proton Synchrotron (PS), which is likewise injected into the Super Proton Synchrotron (SPS) to achieve a proton energy of 450 GeV. Finally the protons are injected into the LHC. Similarly to the previous synchrotrons, the LHC uses dipoles to maintain a constant circular trajectory. Protons must also be compacted for a long run period, so the implementation of beam-focusing quadrupole magnets are a necessity to maintain beamline intensity. The protons collide at the experiments shown in Figure 1.1. The main point of interest for this thesis is at the *ATLAS* (A Toroidal LHC ApparatuS) detector.

1.1.2 The ATLAS Experiment

The ATLAS experiment is a general purpose particle physics experiment at the LHC. The experiment makes use of a 7000 tonne detector 100m below the surface of the earth. This detector sits along the beam-line and is at the center of the proton-proton collision zone generated by the LHC. The ATLAS detector is composed of many modular layers. A charged particle generated from the interaction region will traverse radially outwards through the detector, while its trajectory is bent by the detector's magnetic fields. In general, the design of the detector is such that the highest precision layers are near the radial center, and the lowest precision is in the outermost region. The ATLAS detector is cylindrical by construction, where the *beamline* is situated along the z-axis (refer to figure 1.2). The detector extends outwards perpendicular to the beamline, where a particle from the collision point will traverse through the inner detector, calorimeter, and muon spectrometer respectively.

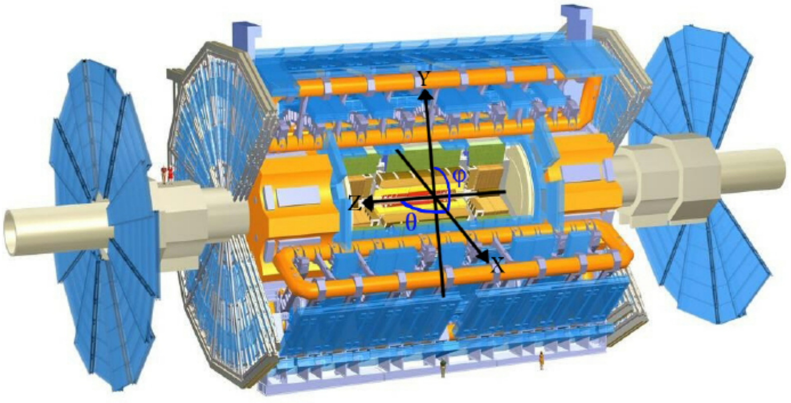


Figure 1.2: The ATLAS detector. The Cartesian coordinate system is overlaid. [4]

1.1.3 Luminosity

The ATLAS experiment at the LHC is looking for physical processes that can be produced by proton-proton collisions. Processes that are rare (have a very low cross-section) will naturally require a large number of collisions to achieve a significant statistical signal from the collected detector data. Many new physics analyses rely on the ability to identify a rare process (small signal) with great precision. Smaller statistical uncertainty can be gathered in two ways; increase the number of proton-proton collisions in the interaction region, or run the LHC for a longer period of time. The former of the choices is selected for increased data collection, because the LHC is already using the majority of the available time to gather data (during running periods).

Luminosity is a key concept that is linked to the underlying motivation for this thesis. Luminosity (L) is a parameter that describes the number of collisions that can occur per unit area per unit time, and is fundamental for increasing the sample size. The value of L can be calculated from:

$$L = \frac{N^2 f}{4\pi\sigma^2} \tag{1.1}$$

where N is the number of protons on the interaction surface, f is the bunch crossing frequency, and σ is the geometric cross section of the beam line. Decreasing the geometric cross section is the method of choice for ongoing luminosity upgrades. This can be achieved with a much stronger aforementioned quadrupole magnet. The initial design luminosity at the LHC is a staggering 10^{34} interactions per cm per second. The interactions at the current luminosity push tracking algorithms to their limits, and an increase in luminosity will continue to challenge the resources of ATLAS.

1.1.4 The High-Luminosity LHC

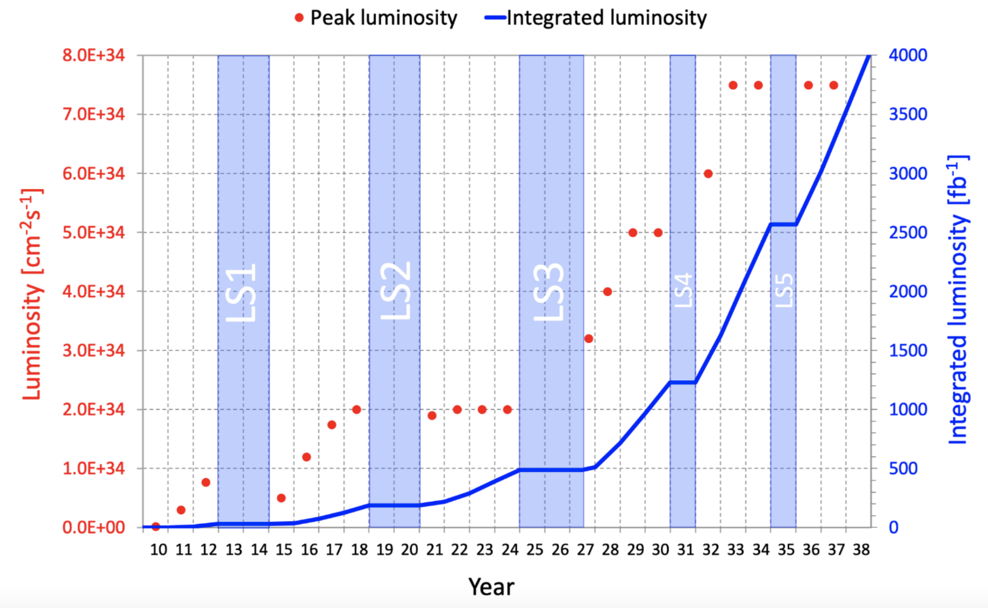


Figure 1.3: Proposed Luminosity increase in HL-LHC. After the long shutdown 3 (LS3 in Figure) in 2026 the expected peak luminosity will increase by up to a factor of 4. [5]

The High-Luminosity LHC (HL-LHC) is the scheduled upgrade to the LHC for 2026, and is the driving motivation behind the project in this thesis. The HL-LHC, among other notable changes, plans to actively gain more proton-proton collisions (increase luminosity) in order to improve statistical uncertainty in measurements. With the improved signal from the increase in data it is possible to provide more precise cross sectional measurements, extend exclusion boundaries for theoretical particles, or perhaps observe something entirely new.

The HL-LHC upgrade implies increasing the proton-proton collision rate, at peak luminosity by a factor of 4 (Figure 1.3). With a bunch spacing of 25 ns, this would yield an order of magnitude increase in interactions per bunch crossings to analyse, relative to the initial LHC design luminosity (from 23 interactions per bunch crossing from the initial LHC design to an expected approximate 200 interactions).

In terms of particle track reconstruction, bunch crossings can be thought of as a snapshot of particle interactions with our detector. Generally the track reconstruction process is more difficult, and computationally intensive when there are more interactions within the detector. This increased computational workload for particle tracking is not the only facet of the ATLAS collaboration that is burdened with the computational requirements (Figure 1.4). For the time being, the overall expected CPU requirements of the HL-HLC exceeds the expected resources from ATLAS from 2026 onwards. There is now a requirement for ATLAS to find less computationally intensive algorithms moving forward. It is the hope that this thesis will provide further insight into quantum computing as an alternative tracking option.

1.1.5 ATLAS Inner Tracker

With the upgrade in proton-proton collision rates, the detectors will also be receiving an upgrade. The Inner Tracker (ITK) is the first line of data collection for the ATLAS detector. It is paramount that the initial trajectories of particles be known. These initial trajectories require the reconstruction of primary and secondary interaction points, known as vertices (e.g. where protons collide). The expectations for the ITK, with current tracking algorithms are: [6]

1. Efficiency for muons greater than 99%.
2. Efficiency for electrons and pions greater than 85%.
3. A rate of fake tracks less than 0.001%.
4. Detector still functional with loss of less than 15% of total channels.

The topology of the ITK is designed to minimize materials, while maximizing the solid angle coverage over the cylindrical detector. This is composed of three parts; the pixel region, end cap regions, and the silicon tracker. The latter two are composed of strips (Figure 1.5). The pixel and strip regions consist of a special silicon doped exterior. This

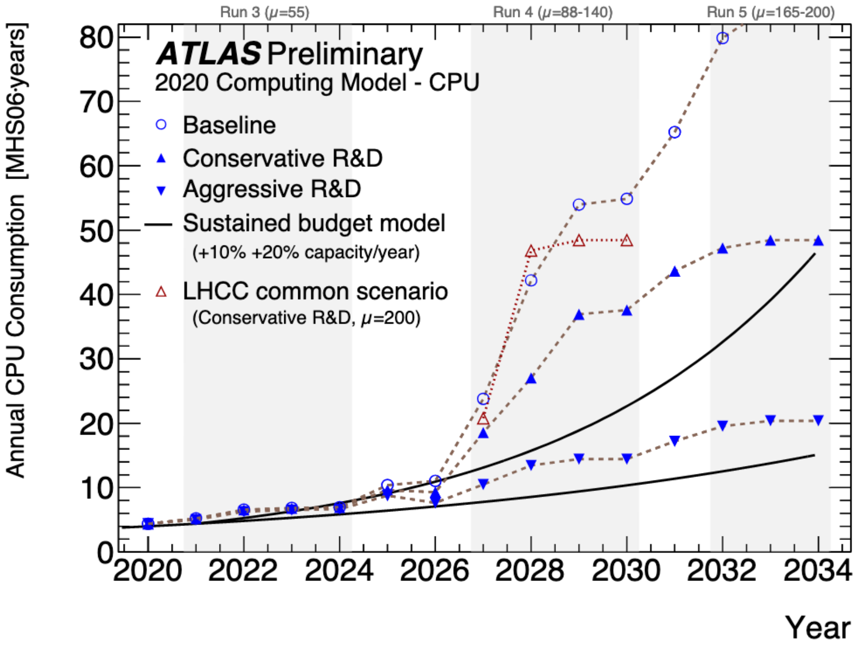
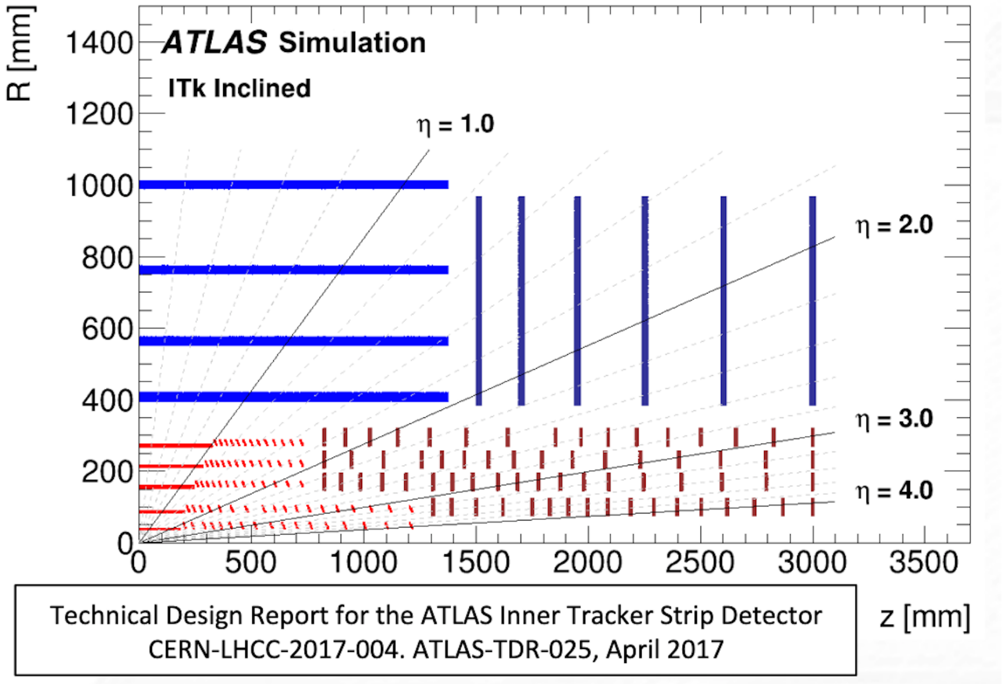


Figure 1.4: Expected computational requirements for the HL-LHC versus a proposed flat-increase computational budget. With aggressive research and development and a 20% increased computational capacity there are sufficient resources, insufficient otherwise.[5]

exterior is reversed biased such that an electron-hole pair can be excited by a charged particle passing through it, this interaction will produce a detectable electric signal which can then be read out by data acquisition machines.

The pixel region is a binary system, either a pixel is activated, indicating a charged particle interacting, or it is not activated. The binary nature of the pixel and strip detectors is a hardware choice designed to make readouts as fast as possible. The activation of a pixel is referred to as a "hit". The pixel detector is the innermost region spanning radially outward to approximately 180mm (depending on the proposed ITK model in use) in a cylindrical coordinate system. The pixel region contains 5×10^9 channels, all of which individually cover a $50 \times 50 \mu m$ or $25 \times 100 \mu m$ portion of the detector. This inner region has the added benefit of being replaceable after an expected lifetime of enduring $2000 fb^{-1}$ total integrated luminosity. The strip regions are also a logic based binary detector. Strip lengths can vary between 18.1 mm and 60.2 mm, and are $320 \mu m$ thick. [7] The strip region extends and maintains a resolution of 23 μm up to a radial distance of 1000 mm (once again, depending on the model).



Technical Design Report for the ATLAS Inner Tracker Strip Detector
CERN-LHCC-2017-004. ATLAS-TDR-025, April 2017

Figure 1.5: The ATLAS ITK topology with Pseudo-rapidity η overlaid. The separation of the detector in the innermost region into small sections is used to minimize the material while maintaining coverage of up to $|\eta| = 4$ [6]

1.2 Tracking

When a charged particle is deflected from the beamline in the LHC, it leaves traces of its presence which can be reverse engineered. Charged particles leaving the detector will be influenced by the ATLAS detectors' magnetic field. This will cause the particles' trajectory to bend based on the momentum of the particle, and its charge. Charged particles will also interact with each layer of the detector that they traverse. The particle interactions with the detector are recorded as "hits" and are later reconstructed.

Tracking is the process of identifying particle interactions within a detector, and combining them into a physical pathway that corresponds to the trajectory of an individual particle. In general, track reconstruction is a complicated version of "connect the dots". The possible dot combinations that can be connected becomes very large (large combinatorics), even when looking at the smallest event sizes. Various techniques will be discussed both here, and in the thesis methodology, to limit this large combinatorial problem.

1.2.1 Metrics in Tracking

In all particle track reconstruction, it is expected that the results will not perfectly match the "truth". For these cases we must score based on two criteria. *Efficiency* (recall) is defined as the ratio of identified true tracks to total true tracks (refer to Figure 1.6). *Precision* (purity) is the ratio of the identified true tracks to total identified tracks. In modern practice, efficiency is the metric of most concern, but this is only because most tracking algorithms have a purity that is close to 1.

1.2.2 Tracking in ATLAS

The ATLAS tracking group runs a sophisticated tracking algorithm with numerous optimisations. With this in mind, the basic ideas of ATLAS tracking are still quite simple, with a fundamental end-goal in mind: find sufficient space points to fully parametrize a particle trajectory in a magnetic field, with as little uncertainty as possible. This is done by numerically calculating the trajectory of a particle in its respective magnetic field (ATLAS does not have a homogeneous field). In order to calculate the full path the particle takes; 5 track parameters are required

$$t = t(d_0, z_0, \phi, \theta, \frac{q}{p}) \tag{1.2}$$

where t is track trajectory which depends on: d_0 the particles' projected closest point to the beamline, z_0 is the distance from the origin to the projection onto the z-axis, and $\frac{q}{p}$ is the charge to momentum ratio (Refer to figure 1.7).

The algorithm for track recombination within ATLAS takes the binary inputs produced by the silicon pixel and strip layers, and turns them into fully parametrized tracks. A simplified graphical version is found in Figure 1.8, it is as follows:

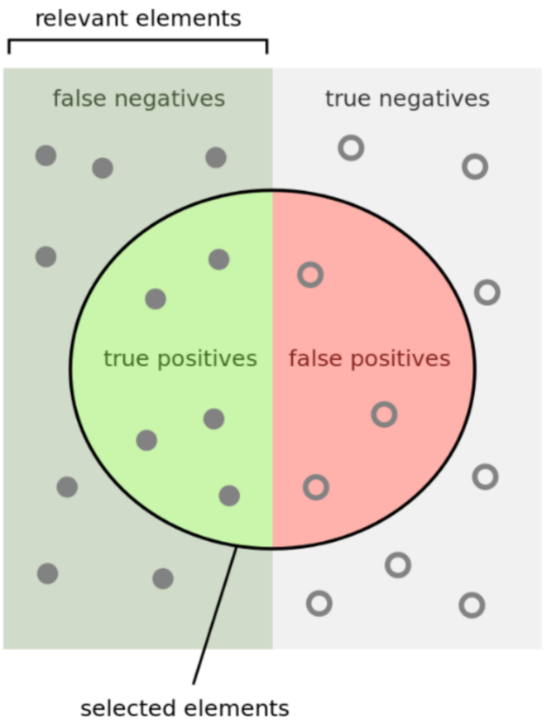


Figure 1.6: Diagram to illustrate True Positives (TP), False Negatives (FN), False Positives(FP), and True Negatives (TN). where Efficiency is the left semicircle (TP) divided by the entire left-most rectangle(TP + FN). Purity is the left semicircle (TP) divided by the entire circle (TP+FP) [8]

1. Initially, proton-proton collisions take place in the interaction region (center of the beamline and detector) and ATLAS records all electromagnetic interactions in the detector layer for 25ns. All of these interactions are recorded as an “event" and placed through track reconstruction [9].

2. Signal output from the pixel and strip regions are simplified into discrete space points. If only a single pixel (or strip) is triggered in local region, then this process is trivial, by claiming that the space-point is the same location as the pixel itself. In many cases, the same particle can trigger many strips or pixels in the same detector layer. If that is the case, one of three things can happen:

   (i) if two or more digital detectors trigger (such as the pixel or strip), the space-point is the boundary between them.

   (ii) if two or more analogue detectors trigger, the space point is determined by the **weighted** midpoint, where the highest weight is given to the detector with the largest electromagnetic signal. This is similar in nature to a centre of mass calculation.

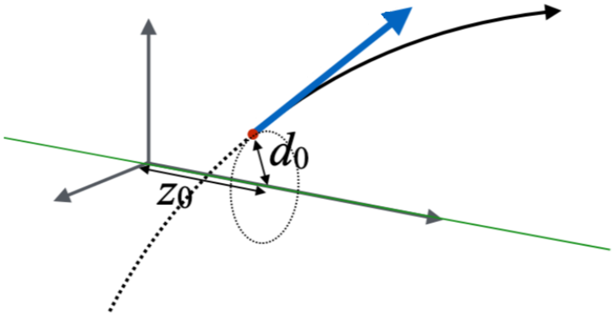


Figure 1.7: Impact parameters used in track parametrisation. (adapted from [9])

(iii) Implement a neural network to use information from several subsequent layers, to guess the true space-point location. This isn't implemented in ITK topologies at the current time.

3. Space-points (referred to as hits in this thesis) must be grouped into potential seeds. Seeds are combinations of 3 space-points that have the potential to extend into a track. In general, a seed is generated if there are 3 hits in subsequent layers, in a similar angular region of the detector. With some exceptions, it is possible for a particle to miss a layer or travel through a dead pixel.

4. Seeds must be extended to track candidates. For bunch crossings of 20 interactions or fewer, generally there are only a few seed extensions available that make any physical sense, given the expected trajectory of the particle. If there are multiple track candidates from extended seeds, the Kalman filter (KF) must be introduced. KF is a recursive linear-quadratic regression which joins a real measurement, and a temporary model prediction, in order to create a more accurate temporary model prediction (to be used with the subsequent real data point). For the purposes of determining track candidates, the algorithm is as follows (Figure 1.9):

   (i) A track seed is extended into a conical region based on its predicted trajectory.

   (ii) If a new space point is anywhere along the trajectory, weight track based on the agreement with the previous predicted trajectory.

   (iii) If there are N separate space points in the cone, and in the same layer, branch into N predictive Kalman filters.

   (iv) If the error of a KF branch is too large, dismiss track candidate.

   (v) Repeat this process until only one track has an acceptable uncertainty (or take the track candidate with the lowest uncertainty if none can be discerned).

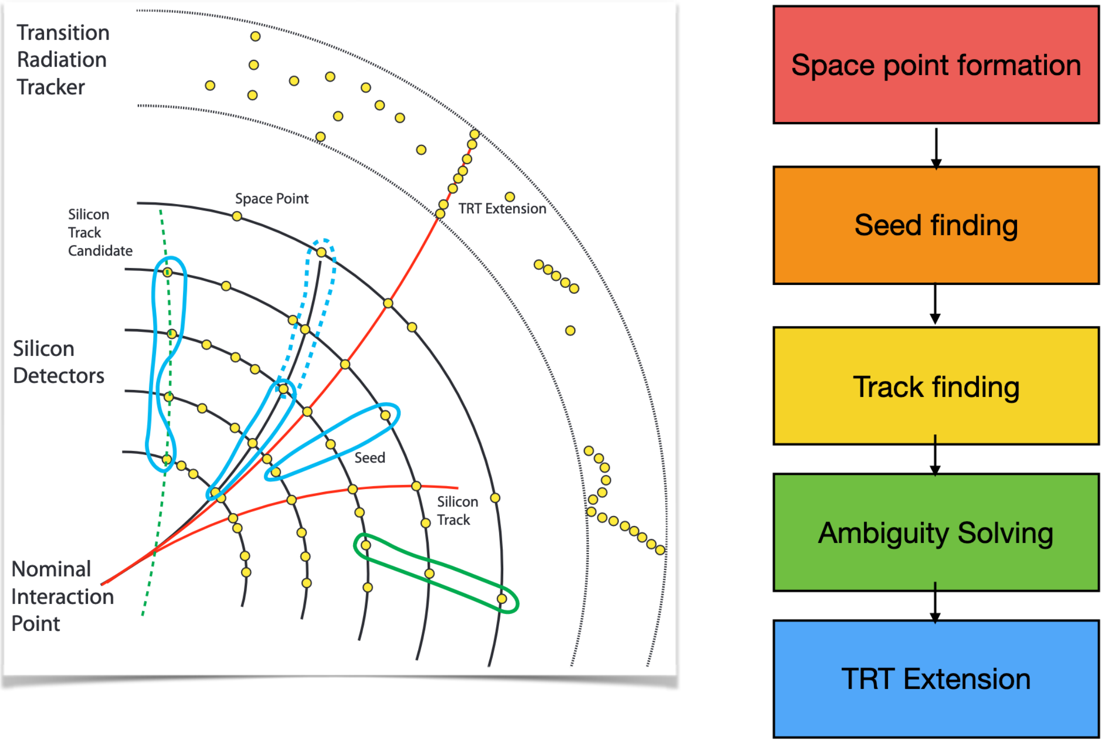


Figure 1.8: Basic stages of ATLAS track reconstruction procedure, taken in the x-y plane.[9]

5. Track candidates sometimes cannot be immediately turned into tracks. It is possible that two track candidates may share a hit, or there are significant gaps in information about a track candidate (many missing layers). Each track is put through a scoring function which applies hard cuts based on: missing hits, quantity of hits in a track candidate, and the effectiveness of least squares regression on track candidate parameters (t). This least squares regression is a minimisation of χ^2

$$\chi^2 = \sum_{meas} \frac{r^2_{meas}}{\sigma^2_{meas}} + \sum_{scat} \frac{\theta^2_{scat}}{\sigma^2_{scat}} + \frac{(sin\theta_{loc})^2 \phi^2_{scat}}{\sigma^2_{scat}} + \sum_{Eloss} \frac{(\Delta E - \overline{\Delta E})^2}{\sigma^2_{Eloss}} \tag{1.3}$$

   where r_{meas} is the difference between the track prediction and the measured track, θ_{scat} and ϕ_{scat} are the differences in angles between the incoming and outgoing track, and ΔE is the energy loss in the material traversed [11].

6. Lastly, remaining tracks are extended farther into the transition radiation detector (TRT) space points. With the inner detector it is now much easier to discern which TRT space point belongs to a track, as the track is already well parametrized.

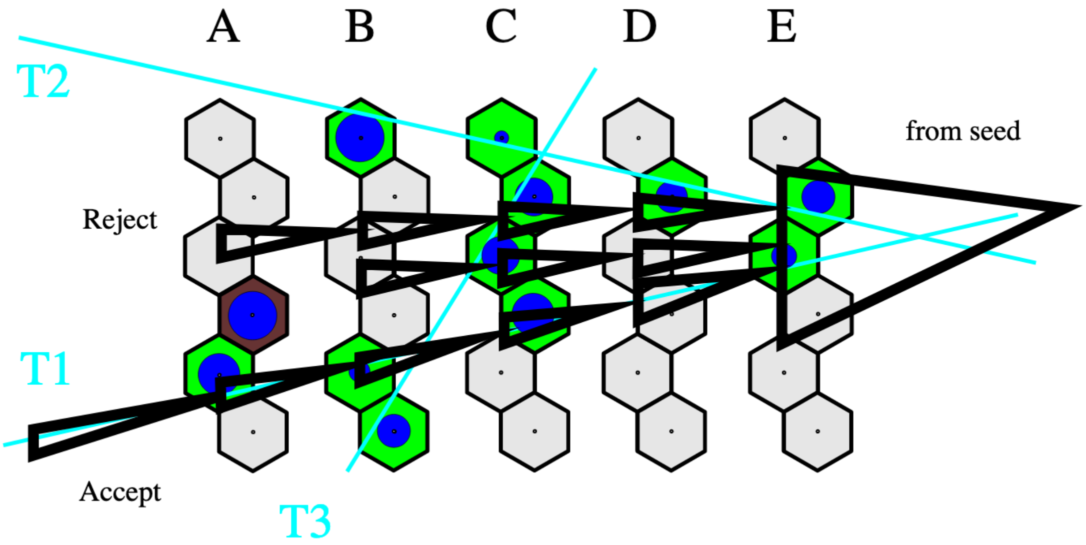


Figure 1.9: Kalman Filtering on probability track trajectories. (The particle pathway is read *right to left*)[10]

1.3 Quantum annealing

Quantum annealing is the new technology that is applied in the track reconstruction algorithm. DWave produces a quantum annealer that is now capable of handling larger problems which can be applied to reconstruction techniques. The quantum annealer has the advantage of performing a single complex calculation which would be resource intensive for a classical computer.

The potential advantage of the quantum annealer is to frame a track reconstruction problem as a single large complex problem, which is easily solved by the quantum annealer in a set amount of time. It is the hope of this thesis to further explore the potential speedup of quantum annealing techniques when compared to the Kalman filter, for large events such as those at the incoming HL-LHC.

1.3.1 Annealing

Problems that require finding a global minimum of a function are NP-hard (non-deterministic polynomial-time hardness). NP-hard class problems cannot be solved in polynomial time. In general, there is no deterministic algorithm that can identify a global minimum of a continuous function. The best that can be done is extensively sampling every region of the function. Annealing is the process in which sampling regions of a function are weighted, depending on the region's rate of *descent* (finding a minimum). Classically, annealing is conducted using a heuristic sampling algorithm.

1.3.2 Adiabatic Evolution

From quantum mechanics, we introduce the idea of adiabatically evolving a system. " A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum." [12] For the purposes of annealing, if a Hamiltonian is in its respective lowest energy level state, and is adiabatically evolved, it will remain in the lowest energy level for the new Hamiltonian:

$$H(S) = A(S)H_d + B(S)H_p \tag{1.4}$$

where H_d is a *driving* Hamiltonian with a well defined ground state that is a transverse magnetic field pointed in the x direction, and H_p is the *problem* Hamiltonian, which we wish to find the ground state of. A(S) is traversed sufficiently slowly from 1 $\rightarrow$ 0, and likewise B(S) from 0 $\rightarrow$ 1. This allows finding the ground state of any complicated H_p .

1.3.3 QUBO Problems

Quadratic Unconstrained Binary Optimisations (QUBO's) are the mathematical framework in which *problem* functions are generated. It is these problem functions that will be adiabatically evolved into the lowest energy state. The problem Hamiltonian can be written as:

$$H_p = -\sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z - \sum_i h_i\sigma_i^z. \tag{1.5}$$

In this function, σ^z are either one or zero (binary), J_{ij} is a coupling coefficient and h_i is a linear weighting coefficient. Both coefficients are set prior to annealing.

In general, creating a QUBO that sufficiently encapsulates a problem is the most difficult part of the annealing process. This involves weighting coefficients effectively, which is non-trivial.

1.3.4 DWave Architecture

The implementation of quantum annealing techniques is currently limited by the available quantum hardware. DWave produces a quantum hardware designed to handle QUBO's. This specialized superconducting hardware is composed of nodes and connectors. A magnetic field can be applied to each node to influence whether its respective quantum state will tend towards a 1 or a 0. This is analogous to the linear weighting term of the QUBO design. Each node has a set number of couplings, which are the physical connections between nodes described by the quadratic terms in the QUBO.

The quantum computer used in this study has a *Chimera* architecture (Figure 1.10), where each node has 6 couplings. Node connections are the first fundamental limitation of the annealing technique. For example; if a QUBO is designed to require 7 couplings on the chimera architecture, 2 nodes must be combined to function as effectively 1 node. This is called a *chain.* In practice this can drastically reduce the effective size of your QPU, and introduces the idea of *problem dependent* computational power in QPU. This thesis will focus on the 2024 Qbit Chimera architecture, which is readily available on the cloud.

1.3.5 Quantum Annealing in Practice

In practice, there are some nuances that come with real-world quantum computing. This is due to living in an era where all quantum computers suffer from noise, or decoherence. In general, the computation is done by completing the following:

1. Configuring coupling connections between nodes.

2. Applying a homogeneous transverse magnetic field (serving as the driving Hamiltonian).

3. Adiabatically introducing new magnetic fields (which is the problem Hamiltonian).

4. Measure the ground state.

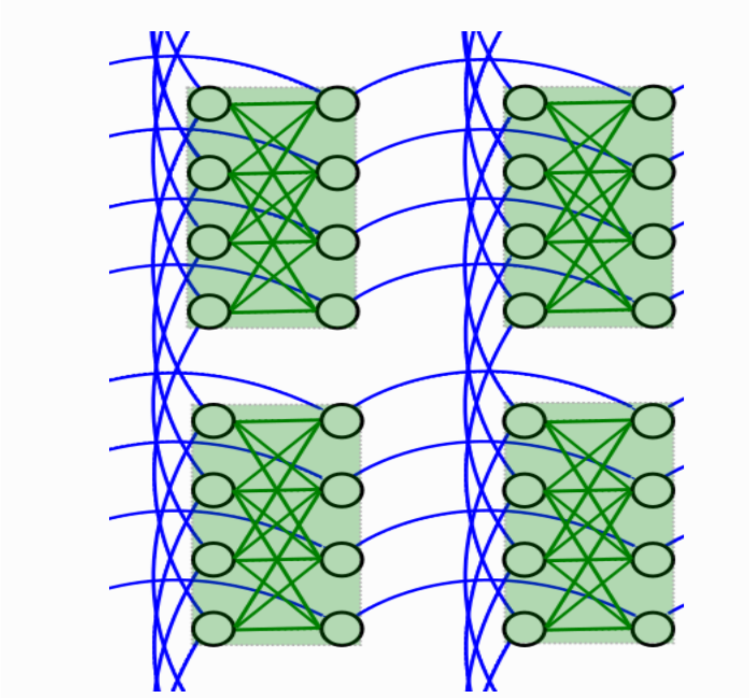


Figure 1.10: DWave Chimera graph. Black ovals are individual nodes. green lines are connections within the cell. Blue are connections to other cells. [13]

5. Repeat until satisfied that the true ground state is measured.

This process takes approximately $21\mu s$ to complete, and serves as a fundamental benchmark for these quantum calculations. In practice due to potential noise, the process of annealing is completed multiple times (the number of total annealing time can be set prior to conserve resources). It is expected that the solution occurring most often is the true ground state.

It is important to note that these DWave annealers are *noisy*. They have a decoherence time on the order of nanoseconds, which is several orders of magnitude smaller than the actual annealing time of 23 μs. This means that the quantum system can jump to an excited state from the intended ground state. This is reconciled by the fact that even with several “jumps”, the most likely end state is still the ground state, or some state that is very close to the ground state energy. Hence the need for multiple annealings for a single calculation.

Chapter 2

Methodology

This chapter will describe the process of transforming raw data into tracks with their associated performance metrics. The raw data must first be converted into a usable form for the algorithm. Then, more complex track-like structures are generated, which are scored based on various hard-criteria. The scores are placed on the purely mathematical framework of a QUBO, which is a usable form for the quantum annealer. Finally, the annealer determines the solution to the track reconstruction, which can then be scored.

2.1 Data Sets

The studies described in this thesis make use of two distinct data sets. The first, where the majority of optimisations are achieved, is the TrackML data. This is followed by the implementation of simulated ATLAS data, for both low and high luminosity. Comparing and contrasting these two data sets allows for some comments about the viability of the quantum annealer in a real ATLAS setup.

2.1.1 TrackML

TrackML was originally intended as an open challenge to machine learning experts, in an attempt to find innovative tools to use in real particle tracking.

The TrackML detector is a generic silicon detector, similar to what can be found in the ATLAS experiment (Figure 2.1). It consists of a cylindrical barrel, composed of layers in the transverse plain. The ends of the detector consist of end-caps, which for the purposes of this reconstruction are not used.

TrackML relies on simplified data, in a file of the form:

hit id	x	y	z	truth	volume id	layer id	module id

where “hit id” is an index, “x”, “y” and “z” are the Cartesian coordinates of the interaction within the simulated detector. In the TrackML set there is some variance of the recorded coordinates and the “true” interaction coordinates, as the hit coordinates are derived from

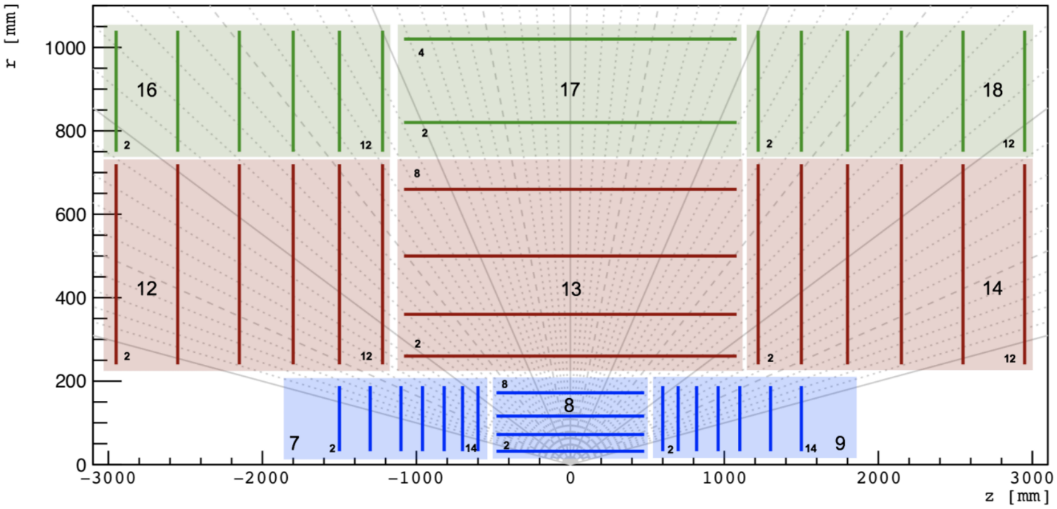


Figure 2.1: TrackML topology in the R-Z plane [14].

cell data within the track topology. “truth” points to the particle that caused the interaction in the detector. “volume id”, “layer id” and “module id” specify the location of the interaction with respect to the topology of the detector. The events generated by TrackML have approximately 10,000 hits, with 10% being “double hits” (a particle interacts twice with the same layer) and 15% of all hits being noise.

TrackML also relies on so-called "truth data" which provides true particle information, without reference to the detector. This truth data is generated during the particle collider simulation that creates the data set itself. This truth data has the form:

hit id	particle id	tx	ty	tz	tpx	tpy	tpz	weight

where “hit id” is the same index as before, “particle id” indicates the truth particle that originates from the center of the detector for the given hit, “tx”, “ty” and “tz” indicate the true interaction coordinates of each hit with a detector layer, “tpx”, “tpy” and “tpz” are the momentum in the x, y, and z direction of the respective truth particle at the interaction point, and “weight” is assigned later as a metric for scoring when assigning a TrackML score. There are also supplementary cell data files and particle files, which for the purposes of the track reconstruction algorithm, are not strictly required .

It is important to note that both the truth file and raw data file have the same number of rows. This is important when trying to convert ATLAS data into this functional format, as the base software package is better equipped to deal with this structure.

2.1.2 ATLAS $t\bar{t}$

The ATLAS simulated data of a top quark and anti-quark pair in the interaction zone of the detector is a product of the ATHENA framework. This particular interaction makes up a problematic high transverse momentum background for searches in new physics. The available dataset has two distinct particle luminosity settings. The first has an expected proton-proton interactions of zero ($\langle \mu \rangle = 0$). This low density is used as a testing measure, when applying the algorithm assumptions used in the TrackML dataset (assumptions such as interaction region at the origin etc.). The second, and more daunting set is $\langle \mu \rangle = 200$, which is similar to real ATLAS experiments at the upcoming HL-LHC. The hit files of a $\langle \mu \rangle$ = 200 event typically have 400,000 hits, two orders of magnitude larger than the TrackML test sets. These exceptionally high luminosity events will be used to test the limits of annealing based track reconstruction.

The ATLAS $t\bar{t}$ simulated data has some differences in data form. The hit file has the form

ID	x	y	z	truth1	truth2	truth3	truth4...

Where “ID” is the hit index, “x”, “y” and “z” are the interaction coordinated with the detector, and “truth1”, “truth2” etc. are the real particles which caused this hit to be recorded. This is a notable difference when compared to the TrackML data set, which only allows for a single particle to register under one hit id.

The ATLAS truth is distinctly different from the TrackML data form.

truth	pdgID	pt	eta	phi	E	charge	x prod	y prod	z prod	d0	z0

Where “truth” is the index of the particle, “pdgID” identifies the type of particle (this is not present in TrackML), “pt” is the transverse momentum, “eta” is the pseudo-rapidity, “E” is the energy of the particle, “charge” is the particle charge, “x prod”, “y prod” and “z prod” are the coordinates of particle origin (this is not the same as the coordinates of particle interaction), and “d0” and “z0” are the impact parameters of the particle. It is important to note that the number of rows in the hit file far exceeds the truth file. In order to get a line by line structure like TrackML, extra pre-processing is required.

2.2 Overview of Algorithm

The following section will describe in more detail the steps taken in this track reconstruction algorithm. The algorithm follows steps that take individual hits, and translate them into full tracks, as well as score them accordingly. A summarised algorithm includes the following steps:

1. Individual hits are turned into pairs of hits (Doublets) using a proximity criteria

2. Doublet structures are extended into 3-hit structures (Triplets) based on preset curvature criteria
3. Triplets are then given connection strengths with surrounding triplets. If two triplets share a doublet segment, these triplet pairs, along with their connection strengths result in a 4-hit structure known as a Quadruplet. The term Quadruplet is most convenient for nomenclature, but should be thought of as a desirable pair of triplets with an associated strength coefficient.
4. All triplets, and their associated connection strengths to surrounding triplets are mapped to a Quadratic Unconstrained Binary Optimisation problem form (QUBO)
5. The QUBO is solved for its respective minimum, which finds the triplets with the most desirable "track-like" characteristics.
6. The retrieved triplets are broken down into their original doublet form, and scored against the truth original data file.

This algorithm was initially proposed by Simpfl-Abele and Garrido, which turned hits into a three hit structure of a triplet [15]. In the previous quantum tracking work by Lucy Linder, these triplets were selected for the nodes of the quantum computer, and given connection strengths to form a usable QUBO for the DWave machine [8]. The triplet was observed to be the smallest track-like structure with a constrained trajectory. This made the triplet the best candidate for usage as a node on the quantum computer. In further sections, this quantum algorithm will be extended and and tuned with further dataset slicing, and parameter optimisations.

2.2.1 Doublet Generation

The first several steps of track generation involve a mix of heuristic, and brute force computational algorithms. The first step in creating any particle track is to generate pairs of connected hits (doublets).

Physical tracks will, in general, follow a set of common sense tests. For example, we expect that a particle interaction in a layer at one end of the detector will not be the same particle in the same layer at the other end of the detector. It is also expected that a hit in an inner layer, will not directly connect to an outer layer, without first traversing through other layers of the detector.

These points are motivation for proximity, and connectivity requirements respectively. The first proximity requirement is met by separating the particle hits into 53 local angular regions with respect to x-y plane (with respect to the angle ϕ). This ensures that doublets are connected within their respective, or adjacent slices. This also provides a significant algorithm speed up.

The second requirement is connectivity. Doublets are generated only when there is a small difference in their respective layers. For example; hits in layers 1 and 5 will not form a doublet, but hits in layers 1 and 2 will. For the purposes of this track reconstruction algorithm, a proxy is used for this metric. The difference in total length of the radius from the origin for both hits is used. This has a tendency to allow more layer skips in the inner barrel region, when compared to the outer regions of the detector.

2.2.2 Triplet Generation

The current state of the algorithm requires more connections than doublets themselves. The algorithm creates a more complex 3-hit structure (triplet) by extending doublets into triplets candidates. This is done by joining two doublets with a shared end. Triplet candidates will naturally have some curvature, as they are either false triplets, or real charged particles travelling through a magnetic field. This fact means that triplet candidates must be filtered by a curvature criteria. This is defined by the Menger curvature, which is a function of the curvature of the unique circle that traverses 3 points (Figure 2.2).

$$c(x, y, z) = \frac{1}{R} \tag{2.1}$$

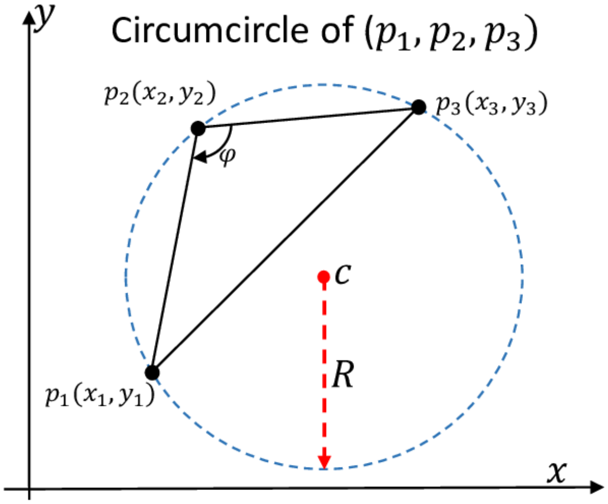


Figure 2.2: The construction of the circle used in the Menger curvature. Where R is the radius of the circle. [16]

The algorithm uses a relaxed Menger curvature requirement when selecting triplets from the candidate pool. The curvature becomes much more important when assessing the quadruplets in the upcoming section. The most crucial filtering requirement for triplets, is that two doublets must extend radially outwards. In other words, the doublet must consist of one hit in an innermost layer, one shared hit, and one hit in an outer layer (Figure 2.3).

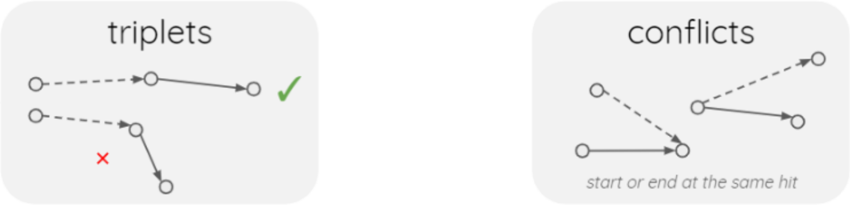


Figure 2.3: Triplets must have an accepted menger curvature (left), and must extend outwards from the center of the detector (right) [8]

2.2.3 Quadruplet Generation

The final requirement before we can begin the annealing process is the final extension of triplets into a quadruplet. Quadruplet candidates are selected from pairs of triplets that have an overlapping segment. Quadruplets must consist of hits that extend outwards from the center of the detector. Triplet candidates that do not have similar curvatures (bottom of figure 2.4) are heavily penalized by the objective function created in the next chapter and are not useful when looking for real physical tracks. These “conflicts” are removed from the quadruplet candidate pool.

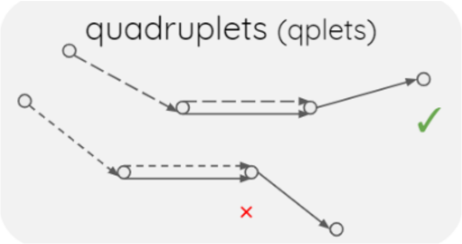


Figure 2.4: Quadruplets must follow a similar curvature, as well as share a segment. In this figure, one triplet is represented by a dashed line, and the other is represented by a solid line. In each case, the two triplets share the middle two hits. [8]

The remaining quadruplets are then scored based on their quality. It is expected that good tracks will have a similar curvature in the x-y plane. It is also expected that good tracks will be straight in the r-z plane. This is because the magnetic field of the detector will not influence the direction within the r-z plane. Finally, the number of missing layer interactions or “holes” should be small relative to the track length. With this, the quadruplet quality criteria is defined as

$$Q(T_i, T_j) = \alpha \frac{\beta(1 - |(\delta(curv_i, curv_j))|) + (1 - \beta)(1 - max(drz_i, drz_j))}{(1 + H_i + H_j)^2} \tag{2.2}$$

where T_i, T_j are the triplets composing the quadruplet, α and β are tunable parameters, where β is critical for balancing the relative importance of particle trajectories in either the x-y, or r-z plane. $\delta(curv_i, curv_j)$ is the difference in Menger curvature beween the particles. drz_i is the difference in angles formed in the r-z plane ($\delta(\arctan(\frac{\delta z}{\delta r})))$. The maximum of this value, from both triplets is taken as the quantity used in quadruplet quality assessment. Finally, H_i, H_j are the number of layer gaps in the respective triplet.

2.2.4 Slicing

It is at this point that triplets and quadruplets can be assigned some “slicing index”. A triplet and quadruplet will have one (or two in an overlapping region) assigned slice(s) (Figure 2.5). This allows for the rest of the algorithm to progress in parallel, but more importantly, is beneficial for the quantum annealing process itself.

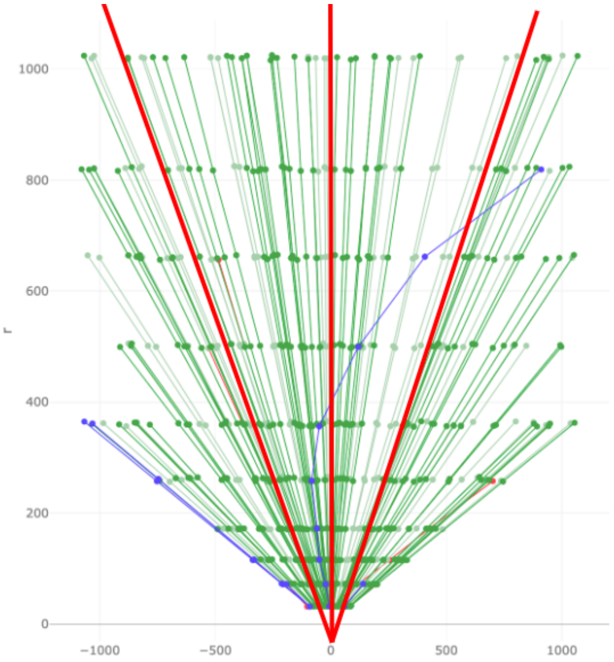


Figure 2.5: Example of 4 slices in the r-z plane (red).

The sliced region overlap is dependent on the maximum allowed triplet and quadruplet drz described in the previous section. This ensures that there are not any dropped quadruplets for the remainder of the algorithm.

2.2.5 Objective Function

The objective function is designed to encode the solution to a problem in its minimum. In the case for the quantum annealer, it must be designed as a QUBO, with the minimum solution encoding all of the physical tracks.

Triplets and quadruplets must be placed into a framework where the most appropriate physical tracks return the lowest result. In other words, the track reconstruction can only be as good as the QUBO design.

The assumptions made in QUBO design are similar in nature to the quadruplet generation itself. Physical tracks probably have similar curvatures as well as minimal gaps in detector layers. This must be put into the framework of linear and quadratic terms, which is the only form usable in the quantum annealer. Thus the QUBO is defined as

$$H(W,S,T) = -\sum_{i<j} -S_{ij}T_iT_j - \sum_i W_iT_i \quad T \in \{0,1\} \tag{2.3}$$

where T_i, T_j are a binary selection of triplets, S_{ij} is the previously calculated quadruplet quality between the two selected triplets, and W_i is the linear weighting term for the individual triplet, which is not required, but can be tuned.

2.2.6 Impact parameters

The linear weighting term in the QUBO can be seen at first glance, as useless. The coefficient can assign either a penalty (positive value) or deem it favorable (negative) to a triplet intrinsically, even if the triplet has no quadruplet connections. This can potentially add more information to the objective function, but must be managed cautiously. Introduced into previous works is the assignment of triplet weights based on perigee impact parameters [8]. These new introductions in conjunction with the base algorithm will be known as the "D0 Model". The base idea of the D0 model is simple; particles that have trajectories tending towards the interaction zone are favorable.

The impact parameters that are applied to the linear weighting coefficient are d0 and z0. d0 is computed by calculating the expected particle trajectory with respect to the beamline:

$$d0 = \sqrt{(cx - ox)^2 + (cy - oy)^2} - cr \tag{2.4}$$

where under the square root is the distance between the z axis and the Menger curvature's closest point of passing. This Menger curvature is parameterized by cx and cy. This particular impact parameter is implemented with caution, as the true trajectory of the triplet is not fully characterized by the Menger curvature. In addition, the magnetic fields in the detector are not necessarily homogeneous, so there should be leniency when extending tracks from the outer layers back into the interaction zone.

z0 is computed by assuming the particle trajectories will fall into the interaction zone along the r-z plane. This is calculated by looking at the two doublets that compose a triplet. If a doublet tends towards the beam center in the r-z plane, it is favorable. The mathematical structure of this is:

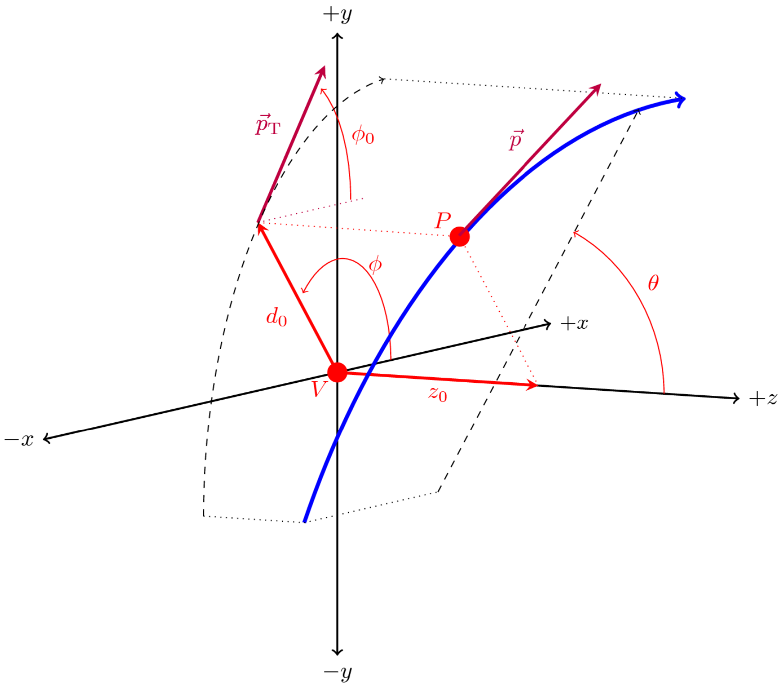


Figure 2.6: Geometry of perigee impact parameters.[17]

$$z0 = cos(\theta_{ab})(max(z_0^{ab}, z_0^{bc}) - bw) \tag{2.5}$$

where $\cos\theta_{ab}$ is the angular difference between the two doublert, z_0^{ab}, z_0^{bc} are both the projected absolute difference between the doublet trajectory, and (0,0) in the r-z plane for each respective doublet ab and bc, and bw is the beam width which is set to 55 mm.

These calculated impact parameters are then put into the linear weighting coefficient:

$$W_i = \alpha(1 - e^{\frac{|d0|}{\gamma}}) + \beta(1 - e^{\frac{|z0|}{\lambda}}) \tag{2.6}$$

where $\alpha, \beta, \gamma, \lambda$ are all free parameters which can be set arbitrarily. It is important to note that previous works showed drastic improvements with these perigee impact parameter weightings, but the particles within the TrackML data set were forced to originate from the origin. This simplification from the TrackML data may not hold, and is one of the main motivations for applying the ATLAS simulated data.

2.2.7 Solving the QUBO

The QUBO (or QUBOs in the case of a pre-sliced triplet set) is now ready to be sent to the quantum annealer. For most datasets, the QUBO is far too large to place on a single quantum annealing architecture. For this, Dwave has a software package which is a classical

- quantum hybrid solver. The idea of this solver is to partition the full QUBO consisting of all the triplet candidates into many sub-QUBOs. In general the number of sub-QUBO's will be approximately 1000 times smaller than the number of triplets. This factor of 1000 is based on the number of required connections (quadruplets) per triplet, and the size of the hardware (in this case approximately 2000 Qbits). These sub-QUBOs are sent individually to the quantum annealer. The sub-QUBO's are selected based on a probabilistic heuristic tabu sampling algorithm (Figure 2.7).

2.2.8 Track Reconstruction

The output of the quantum annealer is only triplets. When plotted, these triplets will take the form of a track. The performance metrics in section 1.2.1 (efficiency and purity) can be taken directly from the doublets that compose the final solution triplets. Track structures can be generated with a recursive connection function. However, this is used in the older TrackML scoring which also required specific hit weightings. This weighting does not work with the current ATLAS simulated data, and no comparison can be drawn from this metric in its current form. For the majority of the results, only efficiency and purity will be compared.

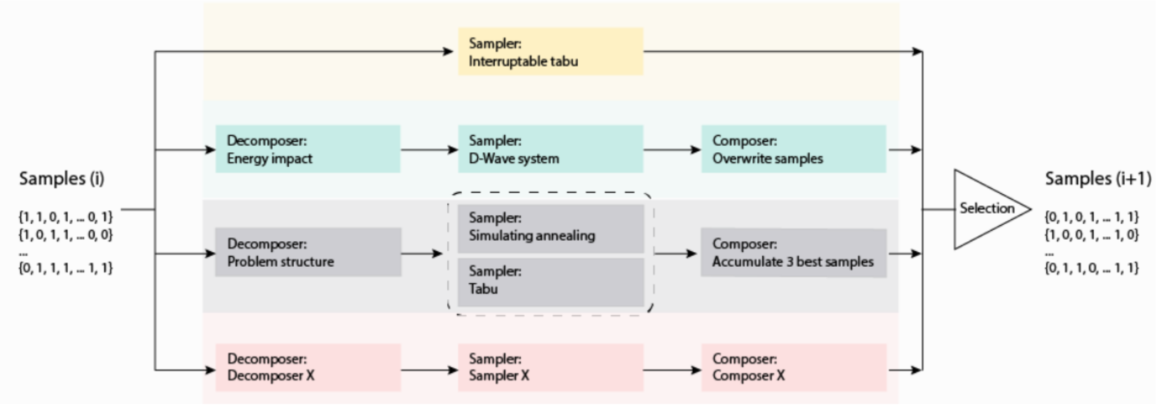


Figure 2.7: Workflow of the classical quantum algorithm. The quantum annealer is dictated by the overarching classical tabu solver [13]

A Note on Classical vs Quantum Annealing

It is important that the relative abundance of classical resources far outweighs quantum resources. DWave provides a classical annealing package which functions very similarly to their quantum annealer. It is their hope (and perhaps expectation) that the quantum annealers will outperform classical annealers in the future. The majority of the work conducted in this thesis relies on the classical annealing software package provided by Dwave. For the remainder of the thesis it will be specified whether the results were obtained with a classical “simulated quantum annealer" or a proper quantum annealer [18].

Chapter 3

Results: TrackML

3.1 Benchmarking Previous Results

The goal of this thesis is the continued development and understanding of the QUBO and its annealing applications in high energy physics. The proof of concept of the base algorithm outlined in the previous chapter has already been established in previous works [8]. This section will display fundamental findings using the algorithm from previous works. These fundamentals include: performance metrics of track reconstruction, the strengths and shortcomings of the base model, the function of both linear weighting term and quadratic connection terms, and the improvements generated by the addition of impact parameters.

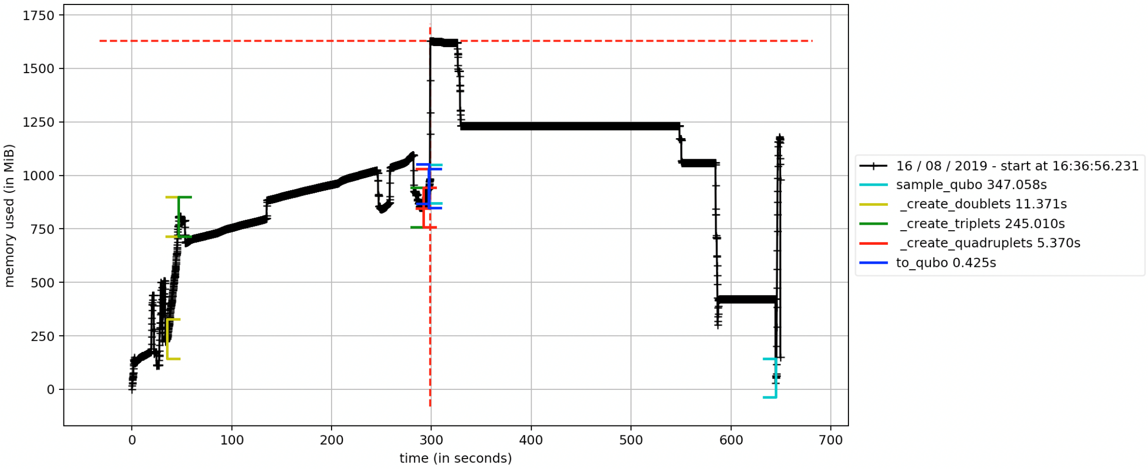


Figure 3.1: Example run of MPROF package within python, allows tracking of start and endpoints of called functions within the package. This particular example shows the progression of memory consumption during the stages of interest in the track reconstruction algorithm (note that to_qubo is just the process of allocating memory to the QUBO in RAM)

3.1.1 Benchmarking: Structure

The preliminary analysis was performed on a TrackML event with 125,000 hits after the initial simplifications of end-cap removal and double hits was performed on a single core of CEDAR (remote supercomputer). Tracks were reconstructed with a given particle density, which ranges from 0.1 - 1, where 1 corresponds to the full event, and 0.1 would correspond to 10% of the truth tracks. The package was profiled with mprof, which allowed for both virtual memory consumption and run time benchmarks for individual partitions of the package (Figure 3.1).

The two consumption metrics tracked by mrpof are the runtime and memory consumption, and these metrics are calculated for two main stages of the algorithm. The two stages are hit processing, and generating/solving the QUBO. Hit processing includes all steps taken to turn raw hits into triplets and quadruplets. Generating and solving the QUBO are the steps turning the objective function into track structures. QUBO performance is the main focus of this algorithm. This QUBO focus is because many particle track reconstruction algorithms already find triplets or “seeds” to initialize longer tracks, and therefore the prepossessing stage isn’t a new stage of development.

3.2 Base Model Benchmarking: Performance Metrics and Consumption

The first model that is tested is the most "basic" of the models. The basic model suffers from overall poorer performance benchmarks than more sophisticated models using impact parameters (see previous section). The QUBO generated by this model is solved on the simulated classical-quantum hybrid solver.

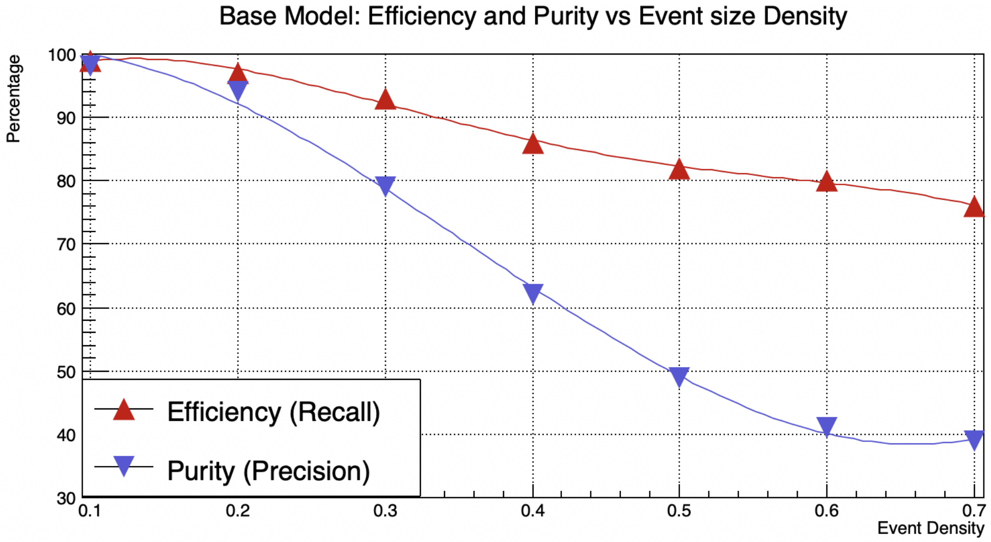


Figure 3.2: QBsolv Benchmarks: Purity and Efficiency of base model

In figure 3.2 The uncertainty caused by the different possible solutions given by the annealer is negligible with respect to the data point size. The efficiency of the base model steadily declines down to 75% at a density of 0.7. The purity decreases dramatically to less than 40% at a density of 0.7. At higher densities there are so few tracks that the probability of a real track turning into a fake track is less than a fake track turning into a different fake track. This drop in purity is significant enough to observe the diminishing returns in purity loss.

Despite poor performance metrics, the base model was still considered because it does not rely on impact parameters that assume the primary vertex is located at (0,0,0). The base model is also computationally non-intensive when quality checking the doublets, as it only requires a single quality pass. It is also noteworthy to reiterate that the base model does not require any linear terms in the QUBO design, and only requires a connection strength. In other words, the QUBO is not deriving any information for track reconstruction from the triplets themselves, but only from the quadruplets they compose.

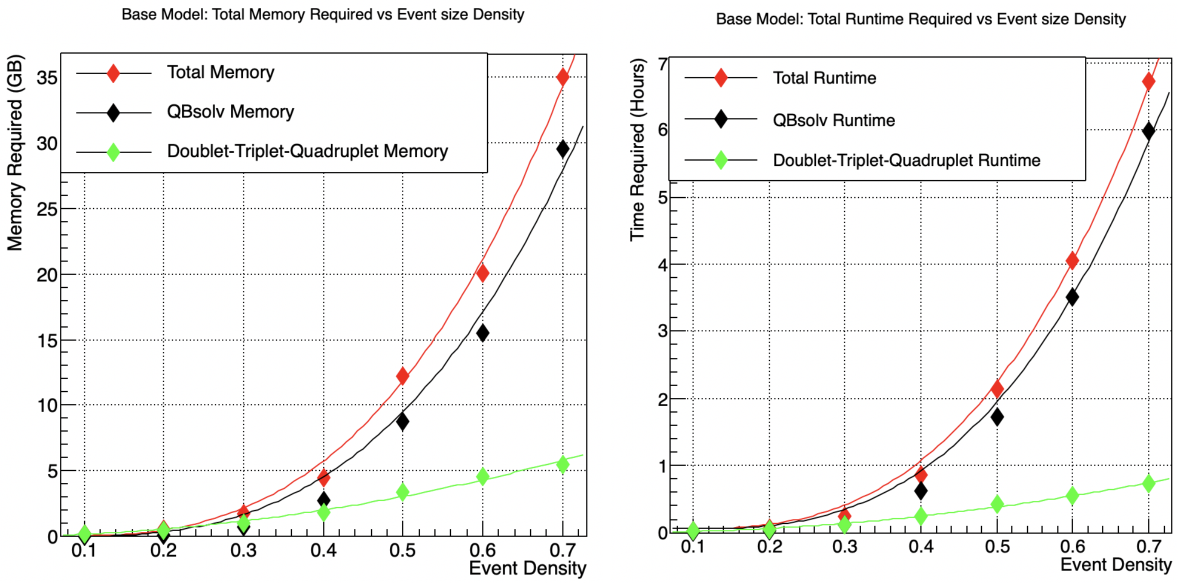


Figure 3.3: QBsolv Benchmarks: memory (left) and runtime (right) for base model

Both memory and runtime behave similarly with respect to the total QBSOLV and pre-QBSOLV (doublet-triplet-quadruplet) consumption. The most notable behaviour is the QBSOLV consumption at higher density events, where both plots are best locally approximated with a 4th order polynomial fit (Figure 3.3 in red). The base model using qbsolv runs into the issue of using significant (> 100GB) memory with a full event size density. This leaves the largest memory and time benchmark at 0.7 density for this specific model. This considerable memory consumption is due to the large space required to retain the large number of sub-QUBO solutions in memory. The large time requirement is dominated by the need to solve the many small sub-QUBOs, rather than the overarching tabu solver. The fundamental requirement for a significant use of time and memory is enough reason to dismiss this model for future studies.

3.3 D0 Model Benchmarking: Performance Metrics and Consumption

The more sophisticated impact parameter model requires a second quality pass on the doublets, as well as additional impact parameter QUBO linear penalty terms defined as:

$$W_i = \alpha(1 - e^{\frac{|d_0|}{\gamma}}) + \beta(1 - e^{\frac{|z_0|}{\lambda}}) \tag{3.1}$$

Where d0 is the distance from the projected curvature of a triplet and the origin, and z0 is constructed using the two doublets composing the triplet from their difference in trajectories along the z-r axis (as they ideally are straight lines towards (0,0)). This was described in more detail in section 2.1.8. The expectation of this model is much superior performance relative to the base model. This is due to the TrackML data being generated near the origin, and the performance bias.

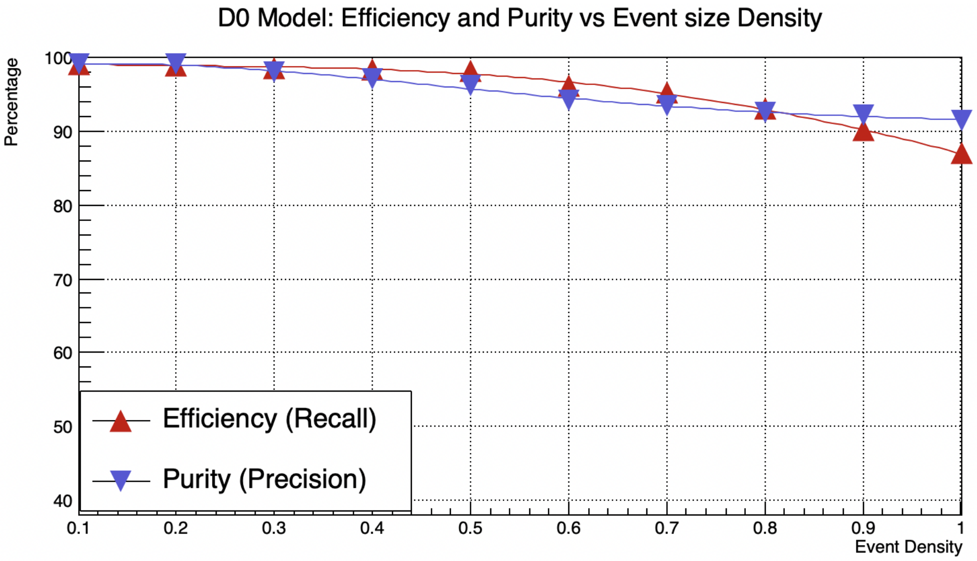


Figure 3.4: QBsolv Benchmarks: Purity and Efficiency of Impact Parameter (D0) model

The impact parameter model performs significantly better with respect to both efficiency and purity (Figure 3.4). It is clear to see the dramatic performance increase with a purity of greater than 90% and efficiency of greater than 85% at the full event density. This leads to the conclusion that deriving additional information from triplets for the QUBO linear terms is effective.

The memory and runtime benchmarks must be inspected to determine the impacts of the second pass, as well as the linear weighting terms on the algorithm. Figure 3.5 displays similar trends for the D0 model when compared to the base model, with some key differences. At density of 0.7 (The maximum benchmarked for the base model) the doublet-triplet-quadruplet generation of the D0 model is 50% more time consuming than the base model. The memory consumption of the doublet-triplet-quadruplet at 0.7 density for both models are comparable within 10%. The pre-QUBO generation delay is insignificant when compared to the time gain of the QUBO solving itself. There is an order of magnitude speed up at 0.7 density for the D0 model. Even with the speed up, QUBO solving comprises the majority of the algorithm runtime in both the base and d0 models. This is a fundamental limitation which is the target of the sliced results section later in this thesis.

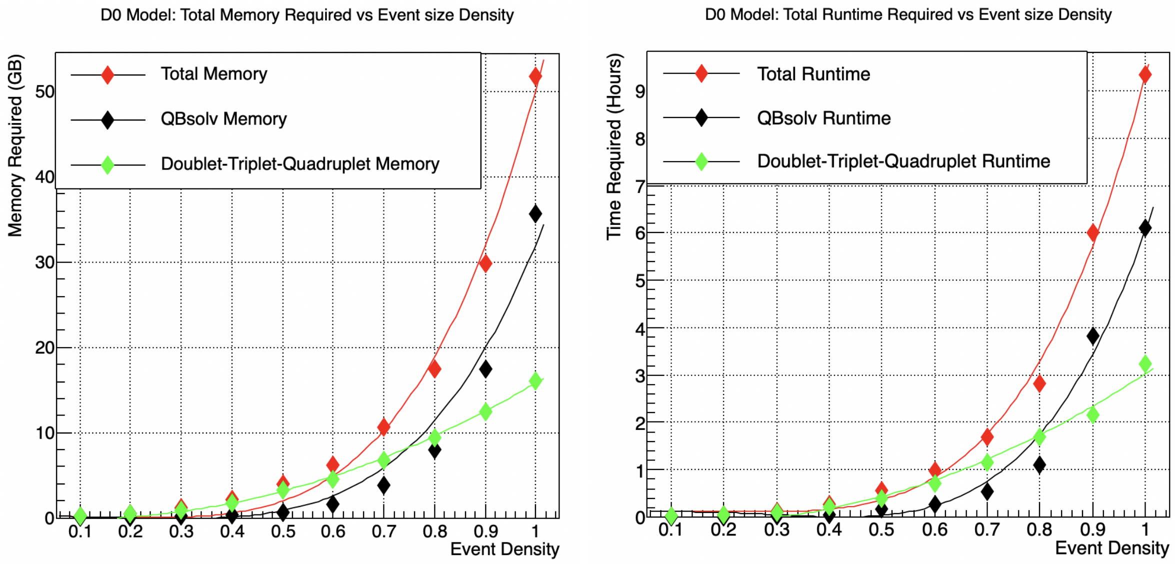


Figure 3.5: QBsolv Benchmarks: memory (left) and runtime (right) for impact parameter (D0) model

3.4 Linear Bias

A natural question can be asked about the role of the linear weighting term in the QUBO structure. Recall that the QUBO has the form:

$$H(W,S,T) = -\sum_{i<j} -S_{ij}T_iT_j - \sum_{i} W_iT_i \quad T \in \{0,1\} \tag{3.2}$$

where W_i is the linear weighting (bias) term for an individual triplet. This term does not rely on any connections to other triplets, and therefore influences the Hamiltonian with the triplet's intrinsic properties. In the base model, this linear W_i term can be set to an arbitrary constant. The plot in Figure 3.6 shows the effect of an constant linear weighting term.

When the weighting term is set to less than zero, it is implying that the solution to QUBO is energetically favourable to have as many triplets as possible that are not in conflict. This loose assumption leads to a low overall purity, but has little effect on the

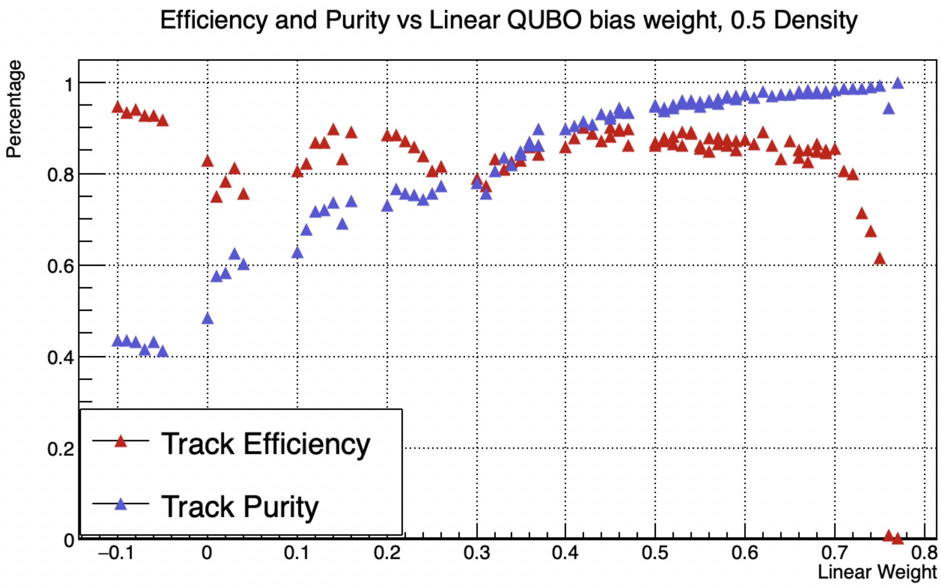


Figure 3.6: Base model 0.5 event density with linear weighting term

efficiency. For large linear weightings, it is clear that there is an improvement in purity with diminishing returns. It is important to notice that for weightings greater than 0.75, the efficiency becomes zero. The sharp drop is due to the weight being sufficiently large that it is energetically favourable for the QUBO to select zero triplets.

When looking at the convolution of both efficiency and purity, the optimal linear weighting is significantly higher than what was used by default in previous works [8] (in this case it's around 0.5 rather than previous uses of 0.2). With this information it is imperative that the linear weighting is an optimization priority when considering all forms of this track reconstruction algorithm.

3.5 D0 model: Energy Solutions

Further considerations are given to the D0 model, as this model relies on a sophisticated combination of linear weighting terms and connection strengths (quadratic terms) in the objective function. It is important that the QUBO design is functioning as expected. The objective function is designed to be optimized such that the lowest energy solution will be the best solution with respect to scoring metrics. However, the retrieved solution is potentially only the best with respect to the objective function itself, and may suffer in scoring metrics. This study attempts to verify if the highest efficiency and purity track reconstructions are being met by the objective function design.

Sampling the quantum annealer involves taking the combination of triplets that supply the minimal amount of energy, as well as recording that associated energy. These retrieved solution energies are then scored based on their difference from the ideal solution energy

(The energy if all tracks had 100% efficiency and purity). Ideally, the difference between the two solutions will be as small as possible.

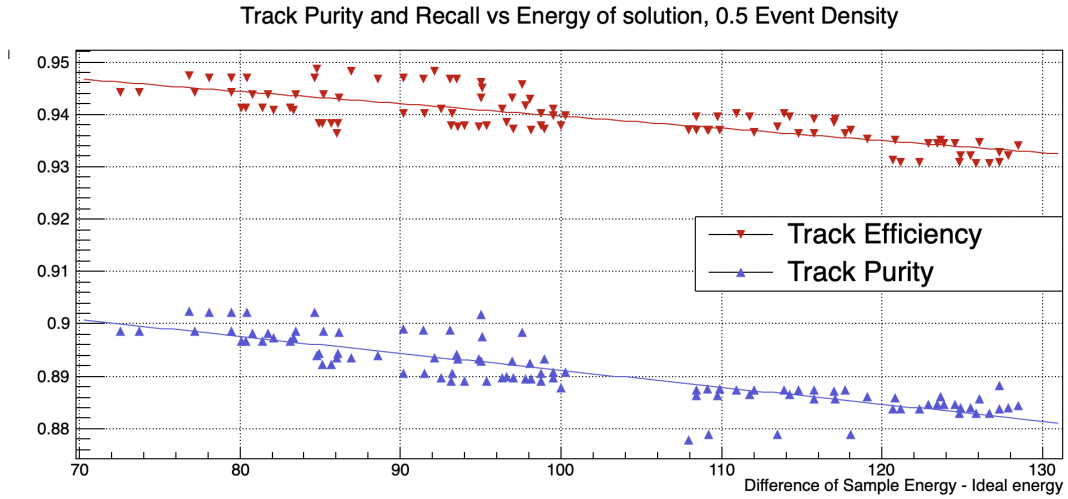


Figure 3.7: Base model 0.5 event density energy solutions with respect to efficiency and purity. The highest rated tracks have the lowest possible difference in sample energy

Figure 3.7 shows this energy difference over many attempts at the same event with a different random seed to generate the 0.5 density event. This study includes lower ranked solutions from the quantum annealer as well (as aforementioned, the annealer is run many times and the best lowest energy solution is selected). The results show both efficiency and purity linearly decreasing with respect to energy difference, as expected. This gives more confidence that there is no underlying fundamental error with the way the objective function is constructed with respect to the linear weight, and verifies that further performance metrics improvements can be derived from the linear terms without issue.

3.6 Impact Parameter Optimisation

The final benchmarking study with the algorithm is the effect of the impact parameter (D0) model with the variation of each respective penalty term. Once again, with the impact parameter model the linear coefficient of the QUBO takes the form:

$$W_i = \alpha(1 - e^{\frac{|d0|}{\gamma}}) + \beta(1 - e^{\frac{|z0|}{\lambda}}) \tag{3.3}$$

where there are 4 free parameters to vary for optimisation. In this study α and β will be varied. It is noteworthy that attempts at finding an optimal λ and γ with a binary search showed either no change, or dramatically detrimental effects on the performance. In addition, the optimal λ and γ were unstable across different event densities. λ and γ are set to 0.5 and 1 respectively for this study, which have been shown to be stable.

The process of this study is a scan of α and β in increments of 0.05 from 0 to 1.4. Performance metrics of purity and efficiency (recall) are determined. This study was performed on an event density of 0.5 on the simulated quantum annealer, which has similar performance comparisons to the proper quantum annealer.

For intuition, the variation of α changes the overall importance of an individual triplet's curvature towards the origin (this is described by the Menger curvature). β changes the importance of the two doublets (composing the triplet) tending towards the origin in the r-z plane.

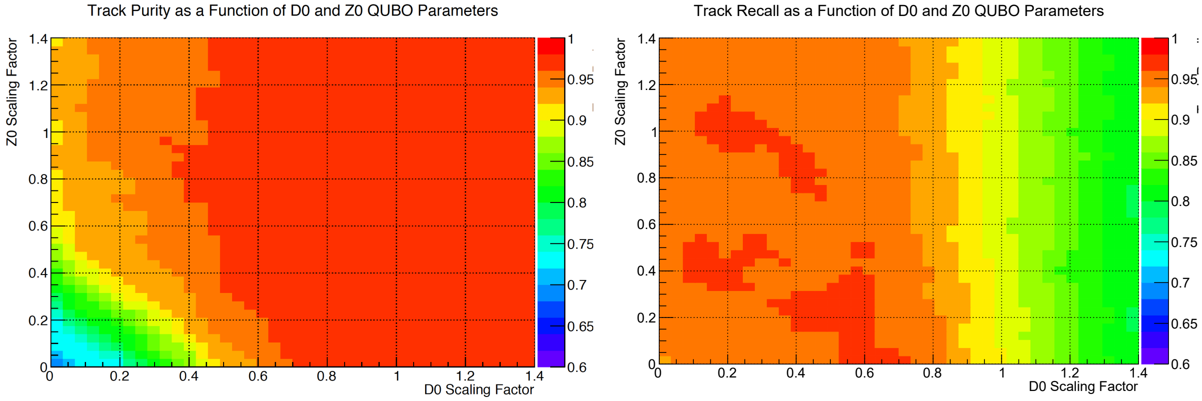


Figure 3.8: Track purity and efficiency with respect to impact parameter penalty parameters α and β for event density 0.5

Figure 3.8 illustrates the effects of z0 and d0 scaling factors on performance metrics. When α and β are set to 0, the performance metrics are very similar to the base model, which is a good cross verification. Purity appears to be a function of both α and β, while efficiency appears to be mostly independent of β in the scanned region.

These metrics are best visualised in convolution with one another. The central region in figure 3.9 shows the optimal region of parameters, but this region is generally large and independent of z0, which intuitively has valuable information to add to the QUBO. This prompted a repeated study at a higher event density of 0.6 seen in figure 3.10. At this density the shape of optimal regions remains similar, however it is clear that there is a much smaller region of preferred z0 parameters. This optimisation shows a 10% increase with respect to the convolution of metrics, when compared to previous studies [8] (where α and β were set to 0.5 and 0.2 respectively with the same λ and γ). For future studies, the default impact parameter coefficients α and β are set to 0.6 and 0.65 respectively.

3.7 Summary

The TrackML dataset, and the robustness of the available quantum annealing algorithm allow for considerable insight into the improvement of quantum track reconstruction. The

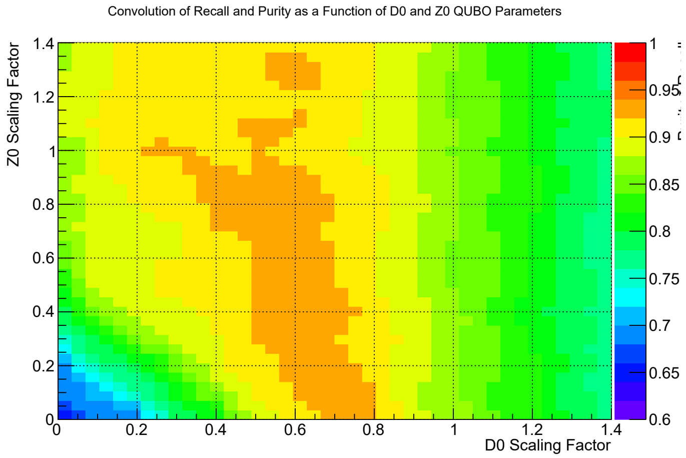


Figure 3.9: Convolution of purity and efficiency with respect to impact parameter penalty parameters α and β for event density 0.5

fundamental value of the linear weighting term is better understood, where it can be tuned to maximize performance metrics within the sophisticated impact parameter model. It is verified that the lowest energy solutions yield the best performance metrics. Most importantly, it has been shown that the fundamental impediment to the algorithm itself is in the consumption metrics. In particular, the time that is required to process a QUBO and retrieve the best solution for larger event densities. This time must be cut down in order to make an argument to use this algorithm in place of an ATLAS default Kalman filter in the future for the HL-LHC. For comparison, the ATLAS track reconstruction for the inner tracker requires approximately 10 seconds to reconstruct an event of $\langle \mu \rangle = 40$ [19]. A proxy for this in the annealing study would be a 70% event density at for the TrackML data set by the number of hits present. It is observed that this annealing algorithm is still 3 orders of magnitude slower than the current ATLAS framework.

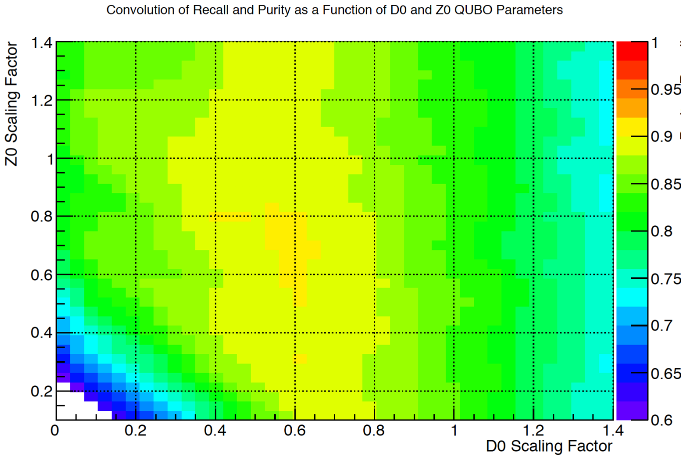


Figure 3.10: Convolution of purity and efficiency with respect to impact parameter penalty parameters α and β for event density 0.6

Chapter 4

Slicing Algorithm

The simplest approach to limiting the QUBO size, is simply to partition the event into various geometrical slices. This partitioning brings the advantage of not requiring any fundamental changes to the algorithm. In theory, it is possible to run the algorithm in N series, where N is the number of designated slices.

4.1 Tuning of Slicing algorithm

The structure of the slices is based on the nature of particle tracks. In general, in the solenoidal magnet such as the one in the ATLAS detector, the particles have some curvature in the x-y plane. This runs into a problem of requiring an extensive overlapping region between slices, such that any valid triplet beginning in a slice can be included into the QUBO framework (See Figure 4.1). The sizing of overlapping regions is non-trivial, as it is expected to encapsulate all reasonable triplets. However an overlap layer sizing that is too large would be detrimental to the algorithm's overall speed, in both triplet generation and overall QUBO size.

The observation that was made early on in development is that the R-Z plane has an ideal linear trajectory when compared to the curved trajectory in the X-Y plane. The slices in the R-Z plane required a significantly smaller overlapping region. It was found that having 4 separate slices in the R-Z plane would help minimize the performance loss of this slicing algorithm.

4.2 Consumption Improvements

The changes in all consumption metrics were significant with the slicing algorithm. QUBO sizes were approximately 25% with respect to the optimised impact parameter unsliced algorithm. Previous observations showed a quadratic-like growth of both QUBO memory and runtime. The quadratic trend is delayed when 4 QUBO's of reduced size are solved sequentially. The total runtime of a full event is 250% larger than this new sliced

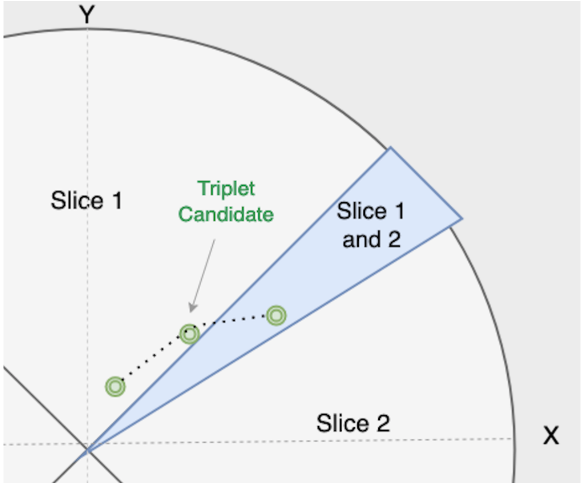


Figure 4.1: Example of a valid triplet that otherwise would be undetected given a slicing scheme without overlap. The overlapping region is defined such that any hit belonging to a region can successfully generate a truth triplet within the overlapping region

algorithm (figure 4.2). However, the total runtime of a full density event is still over 4 hours with these improvements.

Unsurprisingly, a similar reduction occurs with the total memory consumption of the QUBO. At the largest event density, total virtual memory was reduced by a factor of 3 (figure 4.3). The total required virtual memory consumption for a full event required 16Gb of ram. If we consider the total time and space the slicing algorithm requires, it is superior to the unsliced algorithm by a factor of 7 or greater.

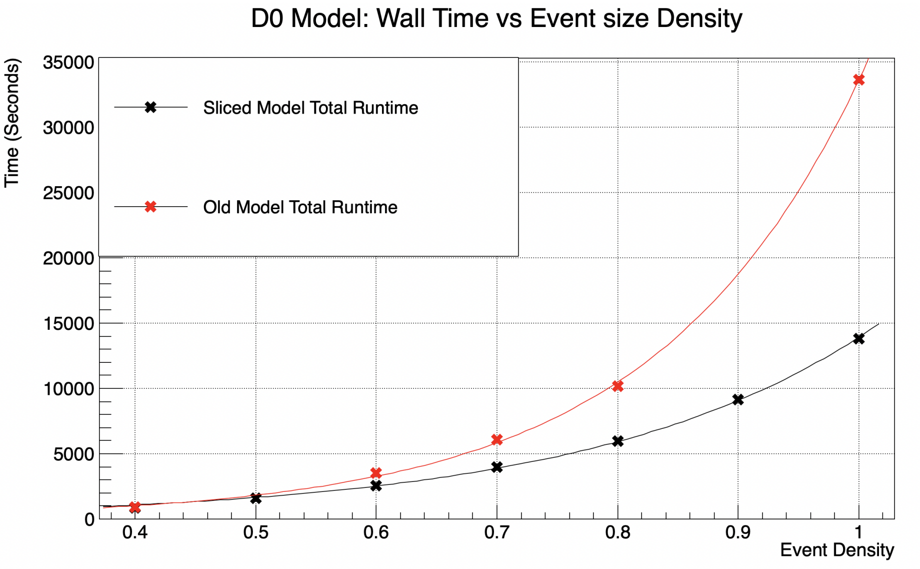


Figure 4.2: Sliced algorithm wall-time consumption: “Old Model" refers to unsliced impact parameter model

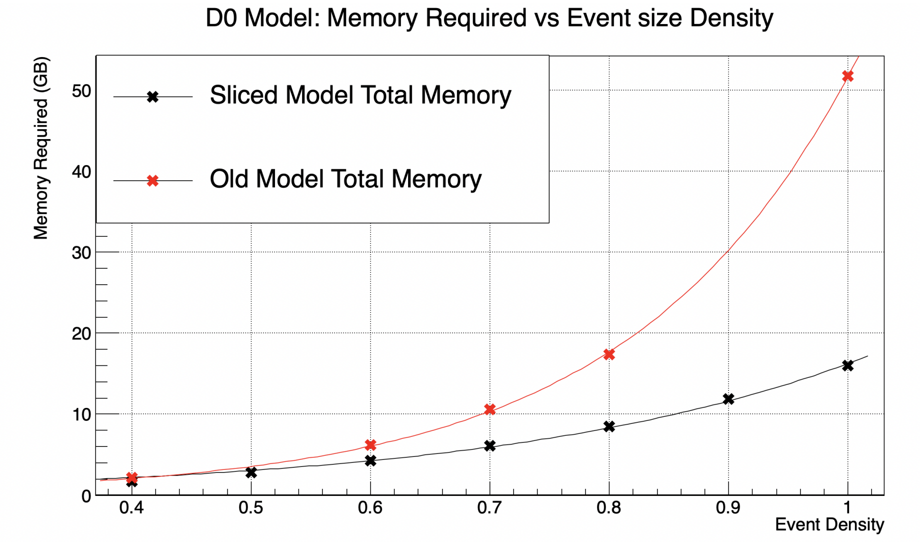


Figure 4.3: Sliced algorithm memory consumption: "Old Model refers to unsliced impact parameter model"

4.3 Performance Metrics

The sliced algorithm in its current form requires a significant performance trade-off to achieve the consumption reduction by a factor of 7 or greater. When comparing purity between the old impact parameter unsliced algorithm, vs the new sliced algorithm, there is a continual decline in performance (figure 4.4). The sliced algorithm suffers a 16% reduced purity at the highest event density, and displays a much greater rate of descent if extrapolated to hypothetically denser events (If we consider upcoming HL-LHC events). The most likely reason for the decline in purity, is the increased probability of creating a fake track. This is due to a triplet being selected that may seem favourable within the scope of a specific slice, but not in the grand scope of the full track reconstruction. This effect is particularly prominent in the slices close to the interaction zone, as the slices are generated with respect to the origin, while particles are not.

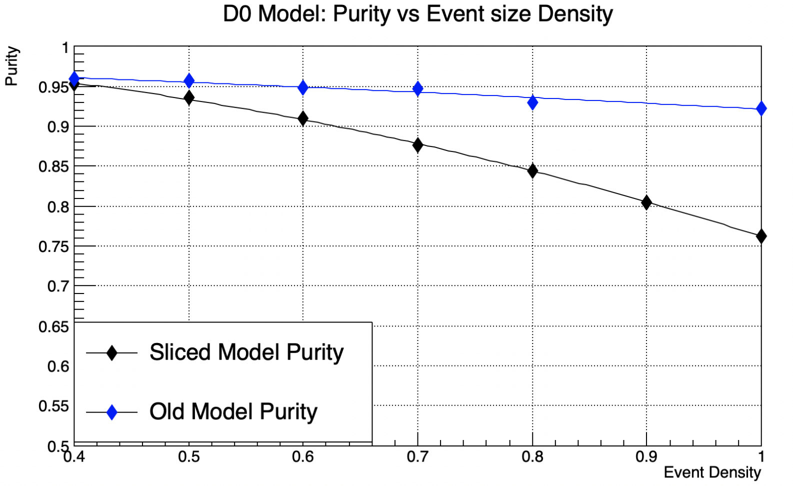


Figure 4.4: Purity

The Efficiency (Recall) of the sliced algorithm suffers from a much less drastic decline in performance (Figure 4.5). However, at a full event density there is still an 8% reduction in efficiency. The current performance drops for the slicing algorithm illustrate a promising reduction in consumption, but the current performance decrease is currently not up to standards for a high accuracy particle reconstruction algorithm.

A natural continuation to the study of slicing is by *how much* the performance metrics deteriorate with respect to the number of slices. As aforementioned, the RZ plane was selected due to a minimal required overlap. In this study, the number of slices is doubled for a maximum of 16 slices. In both Figures 4.6 and 4.7 the total region of overlap is controlled at 10% and 20%. The performance metrics of track reconstruction for many slices, is much worse than a few slices. The downward trend is expected from the nature of slicing with respect to the origin, and not the primary vertex. However, having both

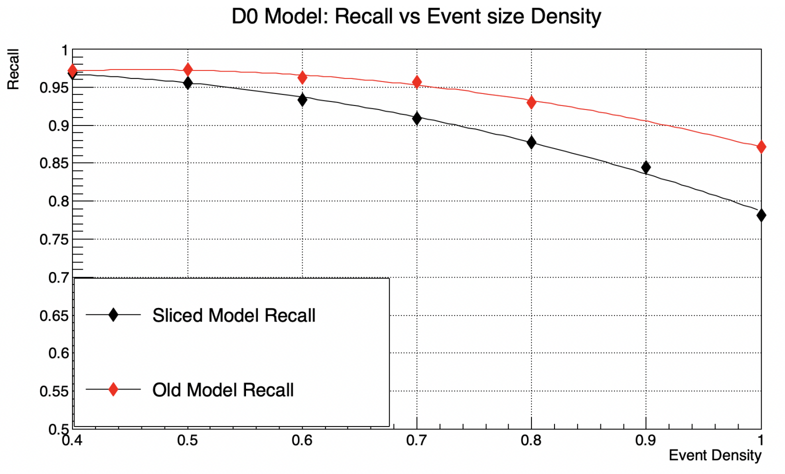


Figure 4.5: Efficiency

efficiency and purity drop below 50% at 16 slices is dramatic. The difference in performance metrics between overlapping regions percentages is much more prominent when there are many slices in the R-Z plane. This difference is greatest when comparing track purity, where an 8% increase can be observed at 16 slices. There is a much less significant performance metrics difference when using a small number of slices with respect to overlap percentage.

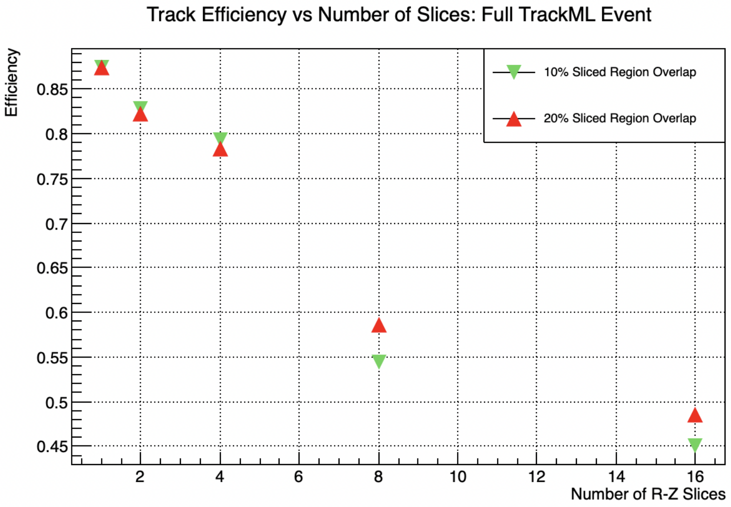


Figure 4.6: Efficiency as a function of slices: Includes both 10% and 20% angular overlap

4.3.1 Size of Quantum Processor With Respect to Slices

This section displayed the promising results of a reduced QUBO size with respect to consumption metrics. The next logical question is to ask: is this a limitation of the algorithm, or the quantum hardware itself? The answer is both. There are methods to process the hits into triplets in parallel, which can give a speedup on the classical computational

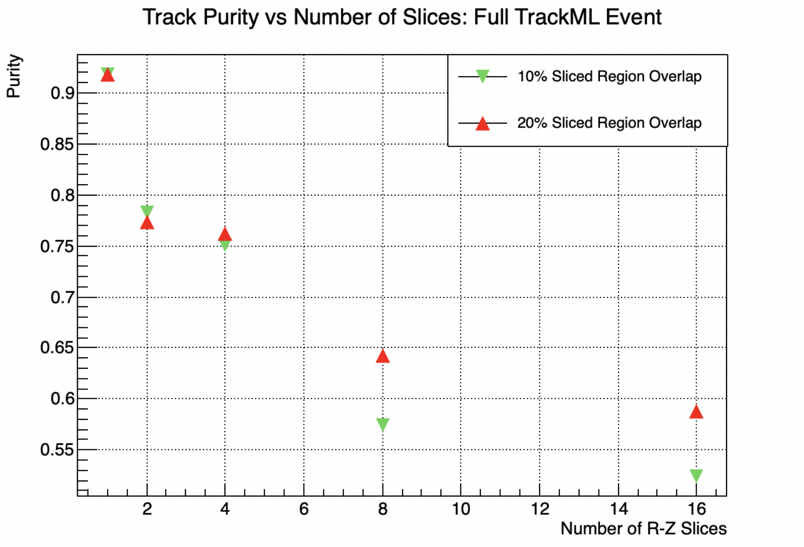


Figure 4.7: Purity as a function of slices: Includes both 10% and 20% angular overlap

portion of the code. The difficulty is managing to reduce the effective size of the QUBO (or number of sub-QUBO's) for Qbsolv to manage. In this thesis the reduction of QUBO size by slicing was attempted, however increasing the size of the QPU is another method for reducing the quantum solving time. This change would effectively remove the downside in performance metrics while maintaining the improvement in consumption metrics. D-wave has since launched a 5000Qbit QPU with a new architecture, which has effectively more than double the capacity of qbits compared to this study, as well as increased inter-connectivity which will facilitate quadruplets better within the algorithm. It is left to further research and development to determine if there is a QPU size limit, at which the QPU can overtake a current sophisticated classical annealing for our given track reconstruction problem.

Chapter 5

Results: ATLAS Simulated Data

This thesis will now explore the new territory of applying the quantum tracking algorithm with an ATLAS simulated dataset. This dataset poses the challenge of determining the robustness of the track reconstruction algorithm when presented with realistic data. This will challenge the previous optimisation assumptions made with the TrackML dataset, and give greater insight into how well the mathematical structure of a QUBO can handle the workload of future real physics events at the HL-LHC.

The insights that will be gained include: the value, and effect of impact parameters (linear weightings), the maximum performance metrics attainable, and most importantly, the most practical performance metrics attainable given a set amount of resources.

5.1 Events: $\langle\mu\rangle = 0$

The available ATLAS $t\bar{t}$ datasets come in two forms: events with $\langle\mu\rangle = 0$ and events with $\langle\mu\rangle = 200$. The initial tests were conducted on the easier to handle $\langle\mu\rangle = 0$ events. This dataset proved significant in establishing a basis for triplet and quadruplet cutting criteria for later higher particle density experiments (See Appendix B). The efficiency and purity of less than 100% on a simple dataset displayed shortcomings with this particular algorithm on ATLAS data at an early stage.

With $\langle\mu\rangle = 0$, $t\bar{t}$ produced an average efficiency of 85% and an average purity of 70%. This is considerably worse than a trackML event with the equivalent density, which would contain both efficiency and purity well above 90%. This benchmark would not change any cutoff values and was explicitly using the TrackML hard cuts. It may be possible to return to this dataset and achieve 100% precision and efficiency with very specific triplet selection criteria in future studies.

5.2 Events: $\langle\mu\rangle = 200$

5.2.1 Simplifications for Dense ATLAS Events

The ATLAS data contained events with an average of 200 interactions per bunch crossing. This translates to approximately 400,000 hits to reconstruct within the detector. This quantity is so large that a considerable time investment of a week is required to run the current algorithm for a single event on qbsolv. Considerable simplifications were required to complete optimisations. The simplifications to the data set are as follows:

1. Hits that do not belong to a particle are discarded (this can be achieved with an immediate comparison to the truth file).

2. For hits that are a result of two particles, one of the particles (and its hits) is discarded entirely.

3. The remaining particles are filtered randomly based on the remaining desired density (50% density would be equivalent to removing half of the remaining particles).

4. A classical annealer (neal) is used to circumvent the extensively long qbsolv time. This gives very similar performance metrics but with substantially reduced consumption metrics.

5.2.2 Optimisation of Performance Metrics: Triplet Cutoffs

An observation that was made early into development of tuning the ATLAS dataset to this algorithm is how previous triplet generation hard-cuts were not functioning as intended, and the performance metrics were greatly suffering at much higher event densities. In general, recall that a triplet is generated if three requirements are met:

1. The two composing doublets have a minimal amount of missing layers

2. Triplets must have less than the maximum allowed Menger curvature. (referred to as X-Y curvature)

3. The two composing doublets must have a similar trajectory with respect to the origin in the RZ plane (referred as the R-Z difference)

Similarly to how the TrackML dataset was probed for impact parameter optimisation, triplet generation requirements 2 and 3 were also scanned.

Referring to figure 5.1, the purity is dependent on the RZ cutoff criteria. The efficiency with respect to the R-Z cutoff is less obvious. Having an RZ cutoff below 0.2 at X-Y curvatures above 0.0014 is shown to be slightly detrimental to the efficiency. The X-Y curvature cutoff was independent of purity, and showed a maximum efficiency around 0.0014.

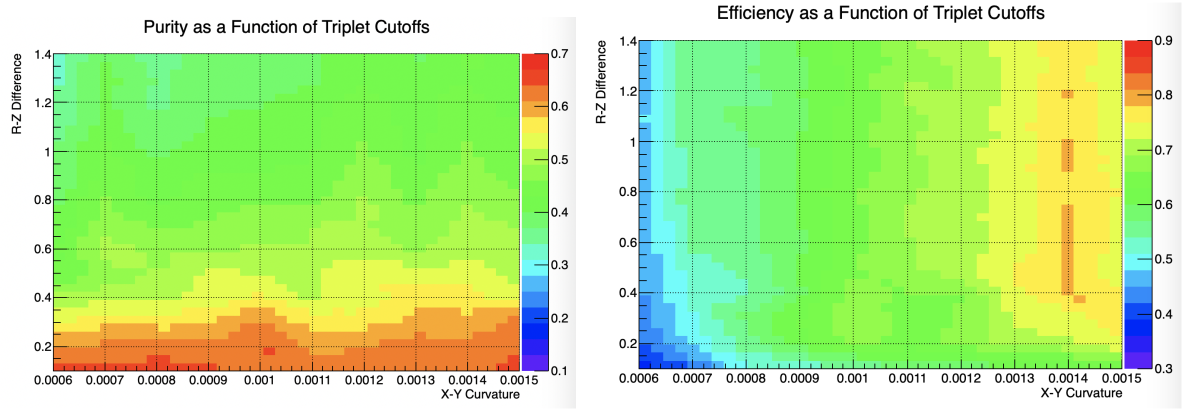


Figure 5.1: Track purity and efficiency with respect to triplet curvature cutoffs within the r-z plane, and the x-y plane.

These results in figure 5.1 showed an optimal performance metrics region along the R-Z cutoff of 0.2, and a X-Y cutoff of 0.0014. In previous studies the default settings were an R-Z cutoff of 0.1, and an X-Y cutoff of 0.0008. These optimal triplet requirements found in this study are almost double the previous acceptance ranges.

Using the new "optimal" triplet cutoff parameters, it was immediately found that the algorithm was using much more resources than previously expected. It is intuitive to think that loosening the cutoff requirements would increase the size of the QUBO. Figure 5.3 displays the growth of triplets generated for the QUBO. Combining this information with the nature of QUBO computation-time growth, it can be seen that loosening the X-Y curvature *drastically* increases resources required to run the algorithm. Going forward, the cutoffs that will be used are 0.2 in the R-Z and 0.001 in the X-Y respectively. These cutoff parameters strike a balance between the convolution of efficiency, purity, and number of triplet candidates generated.

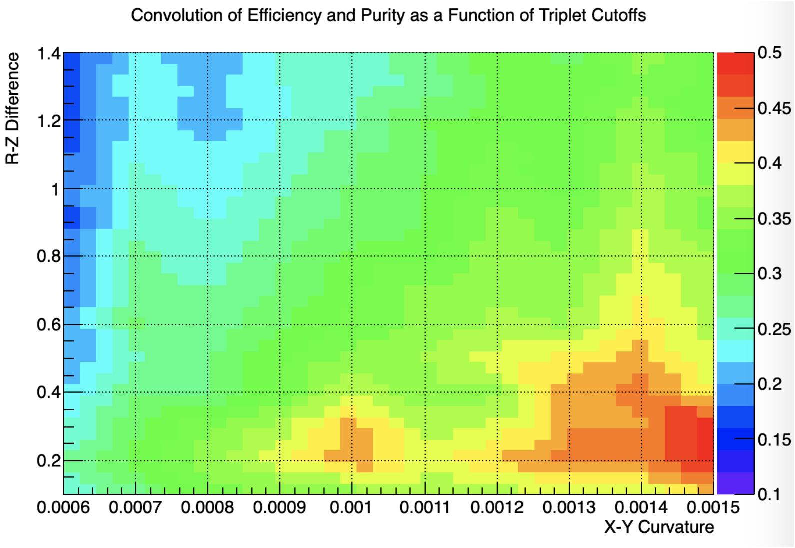


Figure 5.2: Convolution of Purity and Efficiency at event density 0.5: Note the colour axis has a maximum of 0.4

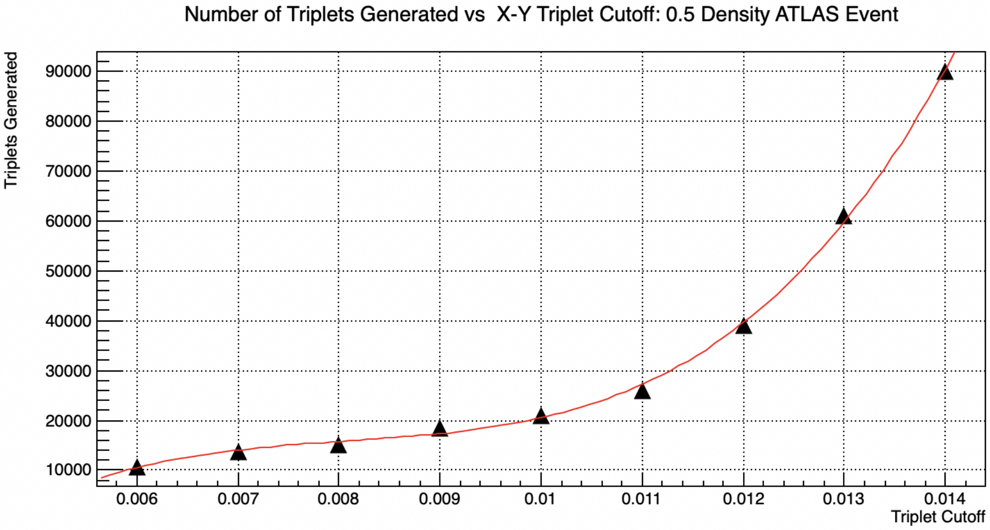


Figure 5.3: Growth of possible triplets within the first pass: R-Z plane cutoff set to 0.2

5.2.3 Optimisation of Performance Metrics: Impact Parameters

A similar study of impact parameter optimisations was performed with the ATLAS larger, more dense dataset. This study scanned a region of linear weighting terms that could be applied to the final QUBO at a 50% event density (figure 5.4). Unlike in the TrackML study, the ATLAS results do not exhibit a clear independence of impact parameters within the scanned region. The reconstruction efficiency was dependent on both d0 and z0 parameters being small, while the purity contained two distinct regions of higher performance. It is important to note how relatively small the optimal values of impact parameter coefficients are (α and β of 0.05 and 0.6 when compared to the TrackML optimal settings of α and β of 0.6 and 0.65). This fact showcases the change in relative importance between the linear weightings (impact parameters) and the quadratic terms (quadruplet connections). The convolution of these two metrics is shown in figure 5.5. Two conclusions can be derived from the result

1. Regardless of the optimisation, the convolution of efficiency and purity does not exceed 0.4, which is significantly worse performance than the TrackML reconstruction

2. It is clear that the usage of the d0 impact parameter is not favourable towards the ATLAS dataset.

If we recall the construction of the d0 impact parameter:

$$d0 = \sqrt{(cx - ox)^2 + (cy - oy)^2} - cr \tag{5.1}$$

the quantity is dependent on the curvature of a triplet, calculated by the Menger curvature. This method was originally suspect, as relying on this curvature does not account for any in-homogeneity in the magnetic field of the detector. In other words, relying on a triplet's Menger trajectory to intersect the interaction zone is not a good measure for particle track reconstruction in a realistic dataset. This fact coincidentally makes the triplet cut relaxation requirement in the previous section more intuitive.

5.2.4 Performance Metrics

In this dataset with $\langle \mu \rangle = 200$, the efficiency and purity decrease as a function of event density (figure 5.6). The rate at which the performance metrics decrease is approximately linear. The effect of consumption metrics was significant enough to limit the study to a 0.8 event density of the $\langle \mu \rangle = 200$ dataset.

At the largest conducted event density of 0.8, the reconstruction efficiency is 52% and the reconstruction purity is 57%. These metrics are well below an acceptable standard for the current ATLAS tracking infrastructure. It is important to note, that in general, the triplet curvature cuts required for the ATLAS dataset were significantly relaxed relative to

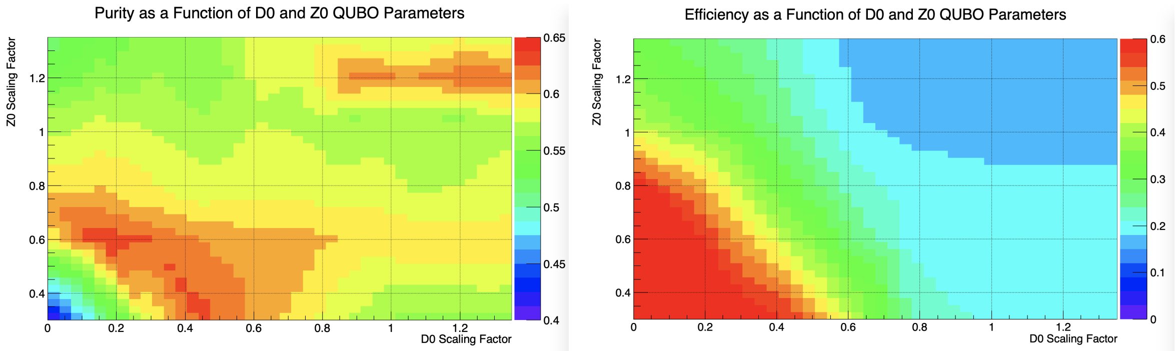


Figure 5.4: Track purity and efficiency with respect to impact parameter penalty parameters α and β for event density 0.5

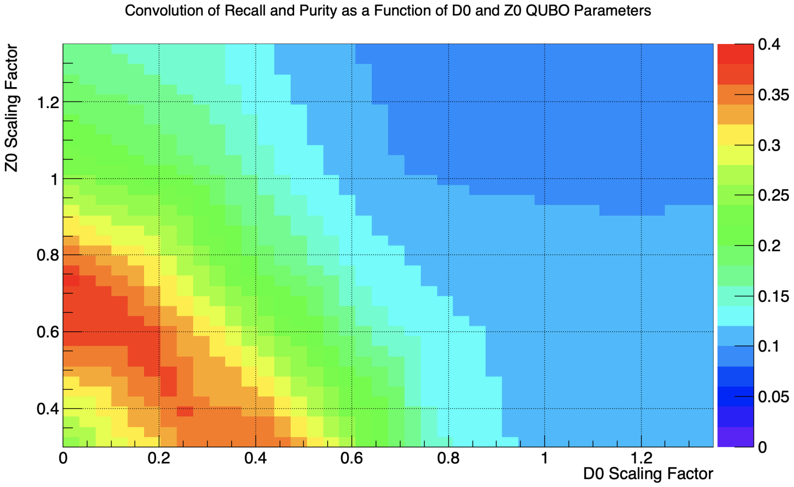


Figure 5.5: Convolution of Purity and Efficiency at event density 0.5: Note the colour axis has a maximum of 0.4

the TrackML criteria. This relaxing was used so that a significantly larger quantity of "real" triplets could be accepted into the QUBO.

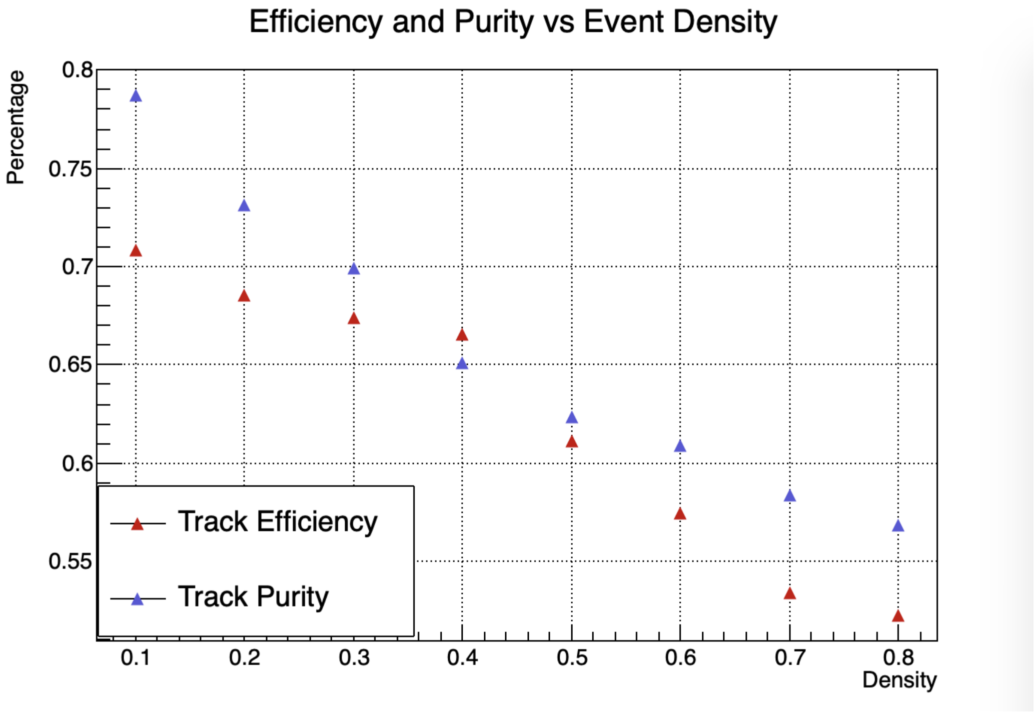


Figure 5.6: Efficiency and Purity as a function of event density for simulated ATLAS $\langle\mu\rangle = 200$ events

Chapter 6

Summary

The goals of this thesis was to further explore the characteristics of the QUBO, and determine the effects of working with a realistic ATLAS dataset. The future for quantum annealing track reconstruction lies in both the speed at which the QUBO can be solved by the quantum computer, and the quality of tracks produced.

The TrackML reconstruction algorithm was optimised with the tuning of impact parameter coefficients. This optimisation led to greatly improved performance metrics within the TrackML dataset. When the TrackML dataset was sliced, it is possible to reduce the growth rate of algorithm consumption, with respect to both runtime and memory consumption. The performance metrics decreased substantially as the number of slices increased.

After several studies with the QUBO on the TrackML dataset, the task of applying realistic ATLAS data to the reconstruction algorithm was undertaken. The process of transforming raw data into a full set of performance metrics is complex, and in the case of ATLAS simulated data, requires a significant amount of simplification. After modifying raw ATLAS data into the algorithm-usable TrackML format, the benchmarks taken were considerably worse than a similar event size in TrackML. This led to multiple studies attempting to reconcile performance metrics. Some minor improvements were made in this respect, but performance remained relatively poor with respect to the previous TrackML dataset.

6.1 Fundamental QUBO observations

One of the main goals of this research project was to gain further insight into the behaviour of the QUBO itself in terms of a track reconstruction problem. The insights made will further the understanding of the long-term outlook of annealing based tracking algorithms.

The TrackML dataset was indispensable as a testing dataset, its robustness allowed for several important (and sometimes uneventful, but important none the less) features of the QUBO to be analysed. These features include:

1. There is a strong correlation between performance metrics and calculated QUBO minimum energies.
2. A small linear weighting term is preferable to none at all, even if that weighting term is a constant.
3. Impact parameter performance within the QUBO is overwhelmingly positive for the toy dataset

When working with a realistic ATLAS dataset, several insights were made with respect to the annealing algorithm:

1. The strict triplet cutoff criteria is insufficient in capturing all truth triplets.
2. The Menger curvature is detrimental when calculating the d0 parameter.
3. The relative importance of quadruplets (quadratic terms in the QUBO) is more than linear weightings in the realistic ATLAS dataset (relative to TrackML).

6.2 Outlook

The long term outlook of quantum annealing in track reconstruction is up for debate. In an ideal environment, the triplets generated in the ATLAS studies within the thesis would be the exact seeds constructed with the sophisticated ATLAS framework. With this in mind, a more qualitative approach must be taken when looking at the outlook for quantum annealing in particle track reconstruction.

As it stands there are several fundamental differences between the quantum annealing algorithm and the current ATLAS standard Kalman filter. The Kalman filter is capable of generating new rules in each new iteration, depending on the needs of the prospective track candidates. These dynamic rules are excellently suited for extending seeds of 3 hits into complete tracks in a noisy, realistic environment. In contrast, the annealing approach is capable of attacking the entire problem in one instance, so long as the hits are turned into triplets (seeds in ATLAS terminology). The drawback is the QUBO includes relatively static rules when applying weights and connection strengths. The current long-term viability of the annealing algorithm will rely on the time-saving from the quantum processor. If hypothetically, there exists a QPU large enough to encapsulate an entire tracking problem, it is possible to convert seeds to tracks within a matter of microseconds. Naturally the QPU technology is currently several orders of magnitude too small to achieve that task. For example: a 0.5 density event for ATLAS $\langle\mu\rangle = 200$ contains approximately 20000 triplets that must be mapped to 2000 active QPU nodes. If it is assumed that there are a sufficient number of connectors within the QPU for each quadruplet, the required QPU must contain 10 times the number of nodes to fit the QUBO without introducing a sub-QUBO solver (for a full density $\langle\mu\rangle = 200$ event this requirement increases by another order of magnitude). It is possible to use the results of the sliced algorithm to estimate the time saving potential of future generation QPU's without the need to fit the QUBO on a single piece of quantum hardware. If the size of the QPU can increase by a small factor, it may be possible to see the algorithm speed benefits of the slicing algorithm without performance drawbacks.

Bibliography

[1] Burkhard Schmidt. "The High-Luminosity upgrade of the LHC: Physics and Technology Challenges for the Accelerator and the Experiments". In: *Journal of Physics: Conference Series* 706 (2016), p. 022002. DOI: 10.1088/1742-6596/706/2/022002. URL: https://doi.org/10.1088/1742-6596/706/2/022002.

[2] The ATLAS Collaboration. *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*. 2012. arXiv: 1207.7214 [hep-ex].

[3] *Public LHC images*. URL: https://lhc-machine-outreach.web.cern.ch/lhc_in_pictures.htm.

[4] Sven-Patrik Hallsjö. "Search for Dark Matter in the Upgraded High Luminosity LHC at CERN: Sensitivity of ATLAS phase II upgrade to dark matter production". PhD thesis. June 2014. DOI: 10.13140/RG.2.1.2837.4481.

[5] P Calafiura et al. *ATLAS HL-LHC Computing Conceptual Design Report*. Tech. rep. Geneva: CERN, 2020. URL: https://cds.cern.ch/record/2729668.

[6] Attilio Andreazza. "ATLAS ITk Pixel Detector Overview". In: (2018). URL: https://cds.cern.ch/record/2652295.

[7] John Stakely Keller. "The ATLAS ITk strip detector system for the High Luminosity LHC upgrade". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 958 (2020). Proceedings of the Vienna Conference on Instrumentation 2019, p. 162053. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2019.04.007. URL: https://www.sciencedirect.com/science/article/pii/S0168900219304553.

[8] Lucy Linder. *Using a quantum Annealer for particle tracking at the LHC*. 2019. URL: "https://github.com/derlin/hepqpr-qallse/blob/master/doc/LucyLinder_TM_2018-2019_S1-online.pdf".

[9] Heather M. Gray. *Track Reconstruction with the ATLAS experiment*. 2015. URL: https://indico.cern.ch/event/504284/contributions/2023875/attachments/1240146/1823137/HGray_Zurich_Tracking.pdf.

[10] S Fleischmann. "Track Reconstruction in the ATLAS Experiment: The Deterministic Annealing Filter". Presented on 19 Oct 2006. 2006. URL: http://cds.cern.ch/record/1014533.

[11] T Cornelissen et al. "The global 2 track fitter in ATLAS". In: *Journal of Physics: Conference Series* 119 (July 2008), p. 032013. DOI: 10.1088/1742-6596/119/3/032013.

[12] M. Born and V. Fock. "Beweis des Adiabatensatzes". In: *Zeitschrift für Physik* 51.3 (1928), pp. 165–180. DOI: 10.1007/BF01343193. URL: https://doi.org/10.1007/BF01343193.

[13] *DWave QPU Architecture: Topologies.* URL: https://docs.dwavesys.com/docs/latest/c_gs_4.html.

[14] *TrackML Particle Tracking Challenge.* URL: https://www.kaggle.com/c/trackml-particle-identification/data.

[15] Georg Stimpfl-Abele and Lluís Garrido. "Fast track finding with neural networks". In: *Computer Physics Communications* 64.1 (1991), pp. 46–56. ISSN: 0010-4655. DOI: https://doi.org/10.1016/0010-4655(91)90048-P. URL: https://www.sciencedirect.com/science/article/pii/001046559190048P.

[16] Zheming Zuo, Jie Li, and Longzhi Yang. "Curvature-Based Sparse Rule Base Generation for Fuzzy Interpolation Using Menger Curvature". In: Jan. 2020, pp. 53–65. ISBN: 978-3-030-29932-3. DOI: 10.1007/978-3-030-29933-0_5.

[17] ""Tikz track" github". In: (2016). URL: https://github.com/ndawe/tikz-track.

[18] DWave. *DWave: Qbsolv Documentation.* URL: https://docs.ocean.dwavesys.com/projects/qbsolv/en/latest/intro.html.

[19] Andreas Salzburger. "Optimisation of the ATLAS Track Reconstruction Software for Run-2". In: *Journal of Physics: Conference Series* 664.7 (2015), p. 072042. DOI: 10.1088/1742-6596/664/7/072042. URL: https://doi.org/10.1088/1742-6596/664/7/072042.

Appendix A

QUBO Generation Parameters

The QUBO parameters within qallse vary with the dataset and study. Included in this section is a description of QUBO parameters, followed by the default values used in studies in this thesis (When running qallse these settings are found in qallse.py and qallse_d0.py).

A.1 Descriptions

max_layer_span: (max_layer_span -1) number of non-sequential layers a doublet triplet or quadruplet can miss without being discarded.
qubo_bias_weight: constant linear weighting term in QUBO.
qubo_conflict_strength: large positive penalty term for poor quadruplet candidates.
num_multiplier: triplet connection strength coefficient.
xy_relative_strength: percentage of importance placed on triplet strength with respect to x-y and r-z plane.
xy_power: x-y plane triplet strength exponent term.
rz_power: r-z plane triplet strength exponent term.
volayer_power: number of layers missing strength exponent term.
strength_bounds: define a maximum allowed triplet strength.
tplet_max_curv: maximum allowed Menger curvature in the x-y plane for a triplet.
tplet_max_drz: maximum difference in doublet slope within the r-z plane.
qplet_max_dcurv: maximum allowd difference in Menger curvature between two triplets.
d0_factor: impact parameter d0 coefficient.
d0_denom: impact parameter d0 exponential coefficient.
z0_factor: impact parameter z0 coefficient.
z0_denom: impact parameter z0 exponential coefficient.
beamspot_width: allowed width of the beamspot in mm.

beamspot_center: coordinates of assumed beamspot center.

A.2 TrackML Dataset Parameters

max_layer_span = 2
qubo_bias_weight = 0
qubo_conflict_strength = 1
num_multiplier = -1
xy_relative_strength = 0.5
xy_power = 1
rz_power = 1
volayer_power = 2
strength_bounds = None
tplet_max_curv = 8E-4
tplet_max_drz = 0.1
qplet_max_dcurv = 1E-4
d0_factor = 0.6
d0_denom = 1.0
z0_factor = 0.65
z0_denom = 0.5
beamspot_width = 55 / 2.0
beamspot_center = (0, 0, 0)

A.3 ATLAS Dataset Parameters: $\langle\mu\rangle = 200$

max_layer_span = 5
qubo_bias_weight = 0
qubo_conflict_strength = 1
num_multiplier = -1
xy_relative_strength = 0.5
xy_power = 1
rz_power = 1
volayer_power = 2
strength_bounds = None
tplet_max_curv = 1E-3
tplet_max_drz = 0.2

```
qplet_max_dcurv = 4E-4
d0_factor = 0.05
d0_denom = 1.0
z0_factor = 0.6
z0_denom = 0.5
beamspot_width = 55 / 2.0
beamspot_center = (0, 0, 0)
```

Appendix B

ATLAS conversion to TrackML form

The following steps are taken to transform the particular ATLAS data into a usable TrackML-like format within the qallse package. Link to the full code is posted to github at `github.com/psreid` under "ATLASread"

1. Generate ATLAS ITK layer structure with respect to the radial distance from the beamline
2. Load in ATLAS truth and hit files into separate dataframes.
3. Remove hits outside the barrel region of ITK.
4. Isolate hits with only one associated truth particle.
5. Apply ATLAS ITK radial layering information to each remaining hit.
6. If the same particle interacts with the same layer, append an extra truth identifier with respect to a loose r-z slicing criteria. If there are still duplicates in the same layer and slice, then remove the truth particle within the slice.
7. Recalculate the number of hits associated with a single truth particle. Append this the respective truth particle dataframe.

Appendix C

Supplementary Plots

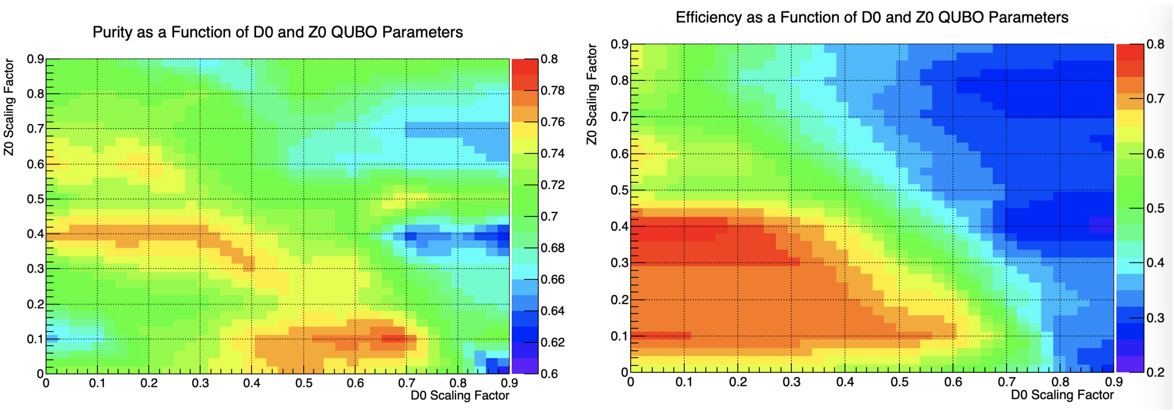


Figure C.1: ATLAS $\langle\mu\rangle = 200$: Track purity and efficiency with respect to impact parameter penalty parameters α and β for event density 0.1

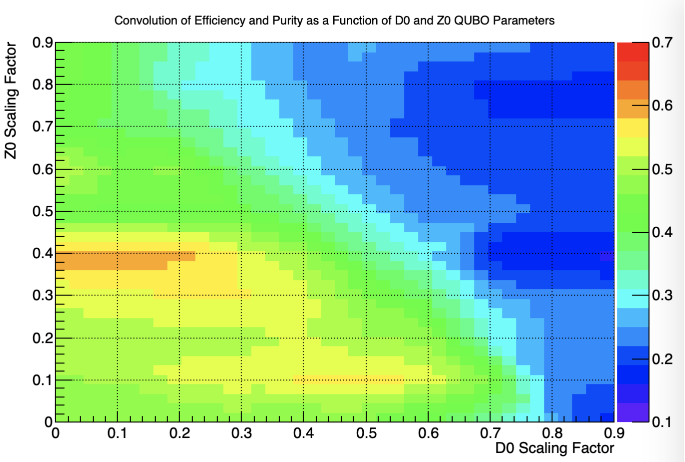


Figure C.2: ATLAS $\langle\mu\rangle = 200$: Convolution of Purity and Efficiency at event density 0.1: Note the colour axis has a maximum of 0.7