SISSA

PAPER • OPEN ACCESS

Improving primary-vertex reconstruction with a minimum-cost lifted multicut graph partitioning algorithm

View the article online for updates and enhancements.

Improving primary-vertex reconstruction with a minimum-cost lifted multicut graph partitioning algorithm

V. Kostyukhin,[a] **M. Keuper,**[b,c] **I. Ibragimov,**[a] **N. Owtscharenko**[a] **and M. Cristinziani**[a,*]

[a]*Center for Particle Physics Siegen, Department Physik, Universität Siegen*
Walter-Flex-Straße 3, 57072 Siegen, Germany

[b]*Department Elektrotechnik und Informatik, Universität Siegen*
Hölderlinstraße 3, 57076 Siegen, Germany

[c]*Max Planck Institute for Informatics,*
Saarland Informatics Campus E1 4, 66123 Saarbrücken, Germany

E-mail: Markus.Cristinziani@uni-siegen.de

ABSTRACT: Particle physics experiments often require the simultaneous reconstruction of many interaction vertices. Usually, this problem is solved by ad hoc heuristic algorithms. We propose a universal approach to address the multiple vertex finding through a principled formulation as a minimum-cost lifted multicut problem. The suggested algorithm is tested in a typical LHC environment with multiple proton-proton interaction vertices. Reconstruction errors caused by the particle detectors complicate the solution and require the introduction of special metrics to assess the vertex-finding performance. We demonstrate that the minimum-cost lifted multicut approach outperforms heuristic algorithms and works well up to the highest vertex multiplicity expected at the LHC.

---

*Corresponding author.

Contents

1   Introduction

In particle physics experiments, many problems require a precise reconstruction of vertices — points in 3D space where particle interactions occur. Knowledge of the positions and features of such vertices provides valuable information about the underlying physics of these interactions. There are numerous examples: *B* physics, heavy-flavour jet identification, primary event vertex reconstruction, search for exotic particles as new physics manifestations, etc. Except for a few specifically designed detectors (emulsions, Wilson chamber, etc.), rarely used in modern experiments, the vertices are not directly detectable. The presence of vertices is usually inferred from 3D traces of charged stable or quasi-stable particles produced in the interaction. Various tracking detectors measure the trajectories of these particles (tracks) in space. The trajectories can be extrapolated to a single 3D point, which represents the interaction vertex position [1].

Despite the simplicity of the vertex reconstruction idea, its real-life exploitation encounters problems. For example, at the Large Hadron Collider (LHC) at the end of Run 2, a typical recorded event may exhibit up to ~80 primary proton-proton interactions, and numerous produced charged particles underwent further interactions leading to additional vertices, distributed in significant 3D

volumes. The expected number of proton-proton interactions in a single event at the LHC after the planned high-luminosity upgrade (HL-LHC) may reach 200–300, resulting in a few thousand reconstructed tracks. The exact amount of vertices in this set of reconstructed tracks is not known a priori. Thus, an efficient inclusive vertex-reconstruction algorithm needs first to determine how many vertices are present in a given track collection, then to assign the reconstructed tracks to these assumed vertices and, finally, to determine the coordinates of each vertex. The track measurement uncertainties, which may differ by a factor of 10 for different tracks and often are comparable with the vertex-vertex distances, cause additional complications. These uncertainties make an exact crossing of track pairs in 3D space impossible: even if two charged particles are produced in the same interaction point, their reconstructed trajectories will only be close to the true vertex position and to each other, up to the corresponding uncertainties.

The explicit reconstruction of multiple vertices from a given track collection can be addressed in a graph-based approach. In fact, all space trajectories of the particles produced in a single vertex should be pairwise compatible, i.e. every pair of tracks should be close to each other in some volume around the true vertex position. Therefore, a compatibility (adjacency) graph can be constructed where every node represents a track. Two nodes are connected by an edge if and only if the distance of the corresponding trajectories is small. In the ideal case, every vertex is represented by a fully connected, isolated subgraph in such a graph. In a realistic scenario, track measurement errors shuffle tracks among different vertices, resulting in a large number of fake edges in the compatibility graph. Yet, it can be tried to partition the full graph into non-overlapping components by cutting some edges so that the remaining edges represent track pairs with minimal distances. Each such graph component determines a union of close-by tracks which can be considered as a vertex approximation.

The present paper focuses on finding primary proton-proton interaction vertices at the LHC. The typical transverse width of the LHC proton beam at the interaction region is $\sim 10\,\mu m$, and the two beams cross each other at a very small angle. As a result, the primary interaction vertices occur in a very narrow and long volume which can be approximated by a line, making this problem effectively one-dimensional. Subsequent decays of the particles produced in the detector volume and their interactions with the detector material, requiring an explicit 3D treatment, will not be considered here. Constraining the vertex positions to a line preserves all important data features while making data generation and comparison with existing approaches much easier.

To illustrate the primary-vertex reconstruction problem, figure 1 shows two zoomed-in regions of a typical simulated LHC event with several pileup interactions.In both plots, the true positions of interaction vertices are shown, together with charged particle trajectories displaced due to reconstruction uncertainties. Several true interaction vertices in these plots do not have associated tracks because all emanated particles in this interaction are outside of the tracking detector's sensitive volume, see section 3 for the details. The overlap of the red (from hard-scatter vertex), blue and grey (from nearby pileup vertices) tracks in the centre of the bottom plot on figure 1 is clearly visible.

Experiments at the LHC use heuristic algorithms [3–5] to reconstruct multiple proton-proton interaction vertices. Several other approaches can be found in the literature, including medical imaging-inspired algorithms [6] and the RAVE package [7] implementing the deterministic annealing algorithm [8]. The latter targets a universal multi-vertex reconstruction, but was not yet used to directly reconstruct primary vertices.
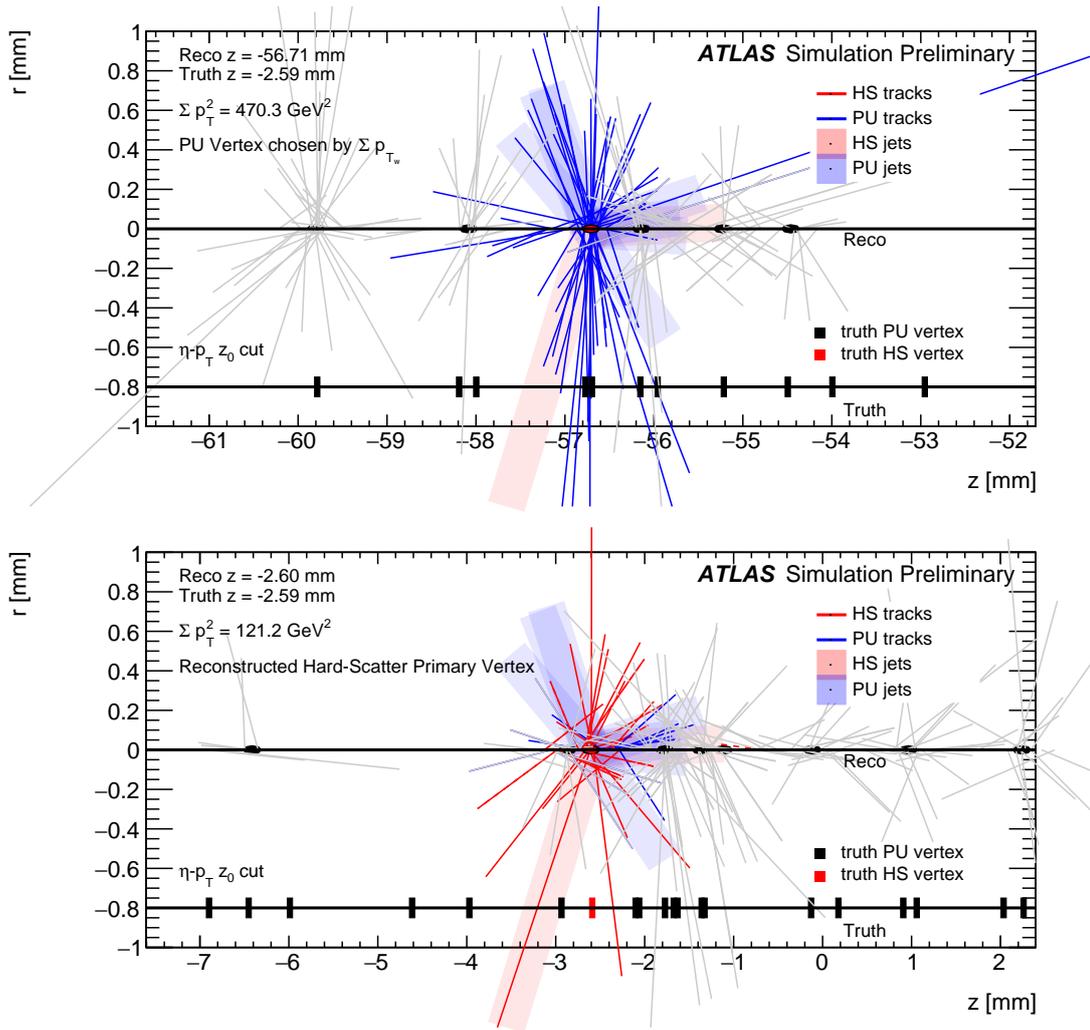
Figure 1. Two regions of a typical LHC event in the ATLAS detector with many pileup interactions [2]. True positions of the proton-proton interactions are shown, as well as the reconstructed trajectories (tracks) of the produced particles scattered due to reconstruction uncertainties. Some truth interaction vertices do not have associated tracks because all emanated particles are outside of the sensitive detector phase space and not reconstructed. These pictures illustrate typical track densities and overlap of the tracks produced in nearby interaction vertices. Both, tracks associated with the hard-scattering (HS) and pileup (PU) are shown.

This article presents an implementation of the Lifted Multicut Graph Partitioning algorithm (LMC), which solves the inclusive vertex reconstruction problem described above. Section 2 describes the LMC algorithm and details of its implementation for the vertex finding application. Section 3 describes the simulated samples which are used to test the algorithm performance. In section 4, features of the simulated samples are discussed. Section 5 introduces edge cost functions used in the graph partitioning. In section 6, the metrics are introduced to estimate the algorithm performance and to compare it with other existing approaches. Section 7 presents the performance of the LMC approach in simulation. In section 8, conclusions are made.

2 Minimum-cost multicuts and lifted multicut algorithm for cluster finding

The compatibility (adjacency) graph representation of the reconstructed track set allows to formulate the primary-vertex finding problem as a minimum-cost lifted multicut graph-partitioning problem. This problem was originally proposed in Reference [9] in the context of image segmentation and mesh decomposition. It is a generalization of the better-known minimum cost multicut problem, also referred to as the weighted correlation clustering problem [10, 11]. The minimum cost multicut problem is a grouping problem defined for a graph $G = (V, E)$ and a cost function $c : E \to \mathbb{R}$ which assigns to all edges $e \in E$ a real-valued cost or reward for being cut. Then, the minimum cost multicut problem is to find a binary edge labelling y according to

$$\min_{y \in \{0,1\}^E} \sum_{e \in E} c_e y_e \tag{2.1}$$

subject to

$$\forall C \in \text{cycles}(G) \quad \forall e \in C : y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'} . \tag{2.2}$$

The constraints on the feasible set of labellings y given in equation (2.2) ensure that the solution of the multicut problem relates one-to-one to the decompositions of graph G, by ensuring for every cycle in G that if an edge is cut within the cycle ($y_e = 1$), so needs to be at least one other. Trivial optimal solutions are avoided by assigning positive (attractive) costs c_e to edges between nodes $v, w \in V$ that likely belong to the same component, while negative (repulsive) costs are assigned to edges that likely belong to different components.

The minimum cost *lifted* multicut problem (LMC) generalizes over the problem defined in equation (2.1)–equation (2.2) by adding a second set of edges that defines additional, potentially long-range costs without altering the set of feasible solutions. It thus defines a second set of edges F between the nodes V of G, resulting in a lifted graph $G' = (V, E \cup F)$, on which we can define a cost function $c' : E \cup F \to \mathbb{R}$. Then, equation (2.1) and equation (2.2) are optimized over all edges in $E \cup F$ and two additional sets of constraints are defined according to [9]

$$\forall v, w \in F \quad \forall P \in v, w - \text{paths}(G) : y_{vw} \leq \sum_{e \in P} y_e \tag{2.3}$$

$$\forall v, w \in F \quad \forall C \in v, w - \text{cuts}(G) : 1 - y_{vw} \leq \sum_{e \in C} (1 - y_e) \tag{2.4}$$

to ensure that the feasible solutions to the LMC problem still relate one-to-one to the decompositions of the original graph G.

For the vertex-finding problem, this formulation allows encoding Euclidean distance constraints in the structure of graph G (e.g. point observations that are spatially distant can not originate from the same vertex), while the cost function can be naturally defined in the distance significance space to take into account the measurement errors. The Euclidean distance and its significance can be very different in case of significant reconstruction errors, the *lifted* multicut formulation encodes both metrics in the same graph.

The minimum cost multicut problem is np-hard, and so is the minimum cost LMC problem [12]. Yet, efficient heuristic solvers provide practically good solutions [9, 13–16]. Here, we resolve to

use the primal feasible heuristic KLj that has been proposed in Reference [9] and published in an open-source library.[1] KLj is an iterative approach that produces a sequence of feasible solutions whose cost decreases monotonically. It takes as input an initial edge labelling (for example, all edge labels are initially set to 0), a lifted graph and costs defined on all edges. In every step, it either moves nodes between two neighbouring components, moves nodes from one component into a new component or joins two components such as to decrease the cost of the multicut maximally according to equation (2.1).

Finally, the compatibility (adjacency) graph partitioning defines the splitting of the initial track collection into a set of clusters consisting of linked tracks with minimal cost. This set of track clusters is a solution of the primary-vertex finding problem, each cluster representing a vertex from which all included tracks emanate.

3   Data simulation

To estimate the track clustering performance based on the compatibility graph partitioning with the LMC algorithm, we simulated data using DELPHES [17]. The framework allows to perform a fast and realistic simulation of a general-purpose collider detector composed of an inner tracker, electromagnetic and hadron calorimeters, and a muon system. For this study, we added a detailed parameterisation of the ATLAS detector tracking resolution to the framework.

To simulate the pileup vertices and hard-scattering events, a sufficiently large amount of minimum-bias interaction events was prepared, consisting of single, double, and non-diffractive processes. These events have been generated using the Pythia 8 [18] event generator. As the main source of hard-scattering interactions, $t\bar{t}$ events are used, also generated with Pythia 8. To simulate an LHC collision event with full pileup, a single $t\bar{t}$ event is mixed with a number of minimum-bias events, distributed according to a Poisson distribution with a mean corresponding to a chosen luminosity. The interaction vertices are then distributed along the LHC beam trajectory inside the detector, according to typical interaction region parameters for ATLAS. Two different descriptions are used: a Gaussian with $\sigma_z = 35$ mm for a collision energy of 13 TeV to be compared ATLAS Run 2 results [4], and a Gaussian with $\sigma_z = 42$ mm for 14 TeV to emulate the HL-LHC environment [19]. Since the LHC beam width is very small, in this simulation it is neglected.

The acceptance of the ATLAS detector allows for reconstructing charged particle trajectories in a limited phase space of $p_\perp > 500$ MeV and $|\eta| < 2.5$. Some minimum-bias proton-proton interactions produce only particles outside the sensitive phase space of the ATLAS detector, which makes them unreconstructable. Positions of interactions with a single track in the ATLAS acceptance can be reconstructed, but this vertex category is contaminated by tracks that are strongly displaced by measurement errors. In the following, a *reconstructable truth vertex* refers to the true position of a proton-proton interaction producing at least two tracks within the ATLAS detector acceptance.

All tracks produced in an event and falling into the sensitive ATLAS detector phase space are smeared according to the parameterised ATLAS detector resolution [20, 21]. Tracks with smeared parameters are referred to as reconstructed tracks in the following. The set of reconstructed tracks corresponding to a full pileup event is used as input for the performance estimation of the clustering

---

[1]https://github.com/bjoern-andres/graph.

algorithms. DELPHES samples used in this paper have been prepared with different energies and different pileup conditions (table 1).

Table 1. DELPHES samples used to estimate the LMC performance. The column $N_{\text{trk}}^{\text{event}}$ reports the total number of reconstructed tracks in simulated events. The last three columns show the average numbers of true vertices with $N_{\text{trk}} = 0, 1, >1$ correspondingly. True vertices with $N_{\text{trk}} > 1$ are considered *reconstructable*.

Energy	$\langle \mu \rangle$	Interaction region σ_z	$\langle N_{\text{trk}}^{\text{event}} \rangle$	$\langle N_{\text{trk=0}}^{\text{vrt}} \rangle$	$\langle N_{\text{trk=1}}^{\text{vrt}} \rangle$	$\langle N_{\text{trk>1}}^{\text{vrt}} \rangle$
13 TeV	63	35 mm	718	9	4	50
14 TeV	150	42 mm	1674	22	9	119
14 TeV	200	42 mm	2227	28	12	160
14 TeV	250	42 mm	2771	35	16	199

4 Features of simulated data

The number of truth tracks in the detector acceptance in the simulated vertices and the position measurement errors of these tracks are shown in figure 2 for a collision energy of 13 TeV. As can be seen in figure 2a, in 14% of the cases, the simulated vertices do not have tracks in the detector acceptance, and in 6.5% of the cases, they have only one track. The number of tracks for all other vertices is widely spread up to 80.
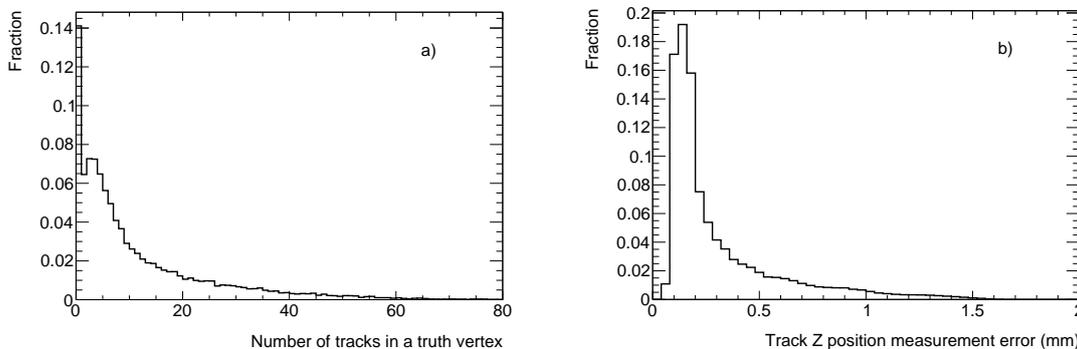
Figure 2. a) Number of tracks per simulated vertex and b) longitudinal position measurement errors for simulated tracks at a collision energy of 13 TeV.

Track measurement errors are shown in figure 2b. From the sizes of the luminous regions and the number of vertices in table 1 we can conclude that the track measurement errors are comparable or larger than a typical vertex-vertex distance in the simulated data. Smearing of the track positions due to measurement errors results in a significant overlap of the tracks from different truth vertices. This effect can be characterised by the truth track overlap fraction, i.e. the fraction of reconstructed tracks in an event that are closer in space to another truth vertex than to the truth vertex of origin. This overlap fraction is shown in table 2 as a function of pileup. An example of the track overlap can be seen in the bottom panel of figure 1. Another example is shown in figure 3.

Table 2. Average truth track overlap fraction in an event, as a function of pileup conditions.

$\langle\mu\rangle$	63	150	200	250
Truth track overlap fraction	20%	41%	53%	66%
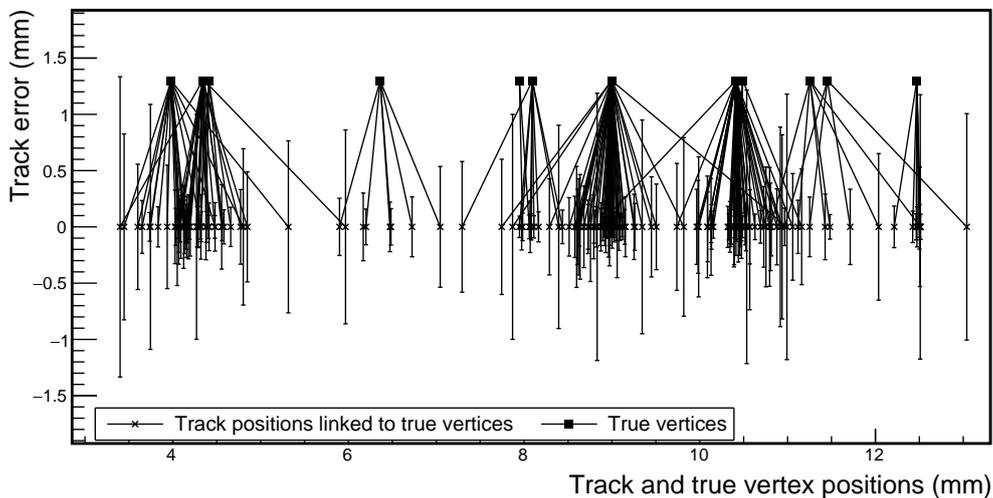
Figure 3. Example display of overlapping tracks from different vertices caused by measurement errors (zoom of a simulated DELPHES event with $\mu = 150$). The crosses at the ordinate value of 0 represent the track positions, and the vertical error bars represent the corresponding position measurement errors. Squares at ordinate values of 1.3 represent the truth vertex positions. The connecting lines show the origin vertex for every track.

A priori, well-measured tracks with small errors should be easy to cluster according to the truth, while poorly measured tracks with large errors can easily migrate from one cluster to another, independently of their true origin. This random migration can be interpreted as noise, and thus, the overall problem may be considered as clustering in the presence of significant noise.

5   Edge weights and constraints

To formulate the vertex finding problem in the presence of pileup as a minimum cost lifted multicut (LMC) graph-partitioning problem, a track-pair compatibility graph needs to be constructed. A node in this graph represents a track, and two nodes are connected by an edge if and only if they are close in space and can be produced in the same vertex. The degree of track closeness, or equivalently the probability of originating from the same vertex, is estimated during the graph construction and is expressed as a weight assigned to the edge. The edge weights determine the efficiency of the partitioning. Therefore, they should incorporate enough information, and the weight assignment procedure should be carefully designed. The following approaches are used in our study:

1. Probability density function (PDF) ratio of the track-track geometrical distance significance based on measured uncertainties, $S = \sqrt{(z_i - z_j)^2 / (\sigma_i^2 + \sigma_j^2)}$;

2. Multivariate binary classification with Boosted Decision Trees (BDT);

3. Logistic regression based on S.

The LMC formulation assumes that the correct edges (two tracks from the same vertex) receive positive weights, while random (fake) edges receive negative weights. This can be achieved by using a logarithm of the ratio of the probability density functions for the correct and fake edges as the cost function of the problem $\log \frac{p_{\text{true}}}{p_{\text{fake}}}$. According to the Neyman-Pearson lemma, this is the most efficient test statistic for the true/fake edge classification. An example of the track-track distance significance distributions and their ratio are shown in figure 4. As the PDF of the fake edges is independent of the track-track distance significance, its overall normalisation depends on the significance range used for the parameterisation. Thus, the exact values of the PDF ratio can be scaled by the choice of the parametrisation range, which in principle, should not affect the LMC clustering performance if the range is sufficiently large. Such a behaviour can be mimicked by a global multiplier of the PDF ratio function. The influence of this multiplier on the clustering will be studied in section 7.3.
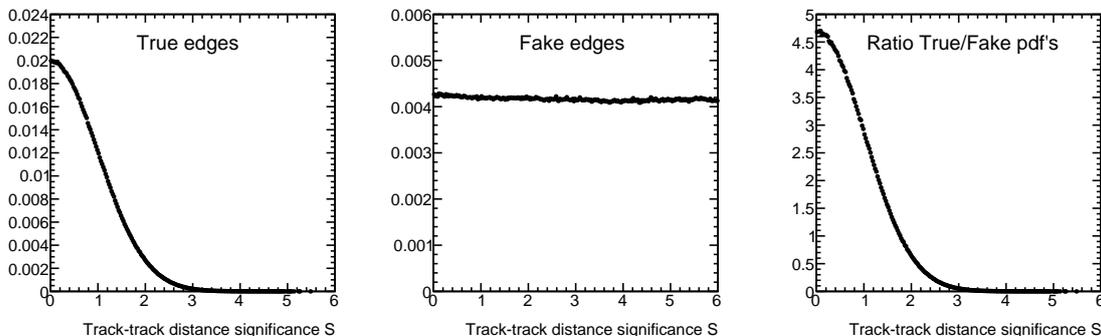
Figure 4. Example track-track distance significance for true and fake edges and their ratio. The significance distributions are normalized to one.

A better clustering performance could be achieved by encoding more information in the edge weight calculation. To test this approach, we use a BDT classifier combining seven features, listed in table 3, to distinguish true edges from fake ones. The GradientBoost implementation (BDTG) from the TMVA [22] package is used to train the classifier. An example of the trained classifier response[2] is shown in figure 5. The output is negative for fake edges and positive for true ones, exactly as required by the KLj algorithm, and therefore can be used directly as the edge weight.

Edge weights can also be assigned by using the logistic regression $p = e^z/(e^z + 1)$, where $z = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$ and x_i are explanatory variables. The negative inverse of the logistic function, $\text{logit}(p) = \log[p/(1 - p)]$, provides the necessary edge weight behaviour. Edges that need to be removed receive negative weights, and those that need to be preserved receive positive weights. The intercept value β_0 is defined by the ratio between the amount of true and fake edges used for training, which can be linked to a prior probability of a given edge being true or fake. In the current problem, the prior probability depends on the true vertex density and cannot be defined unambiguously, e.g. it

---

[2]TMVA GradientBoost uses the binomial log-likelihood loss $L(F, y) = \ln[1 + \exp(-2F(x)y)]$ with Gini Index separation. We use the following training settings NTree = 800, MaxDepth = 10, MinNodeSize = 1.5%, Shrinkage = 0.07.

Table 3. Input features for the edge classification BDT.

n.	Description
1	Squared significance S^2 (or χ^2) of track-track distance along beamline
2	Average position of the track pair along beamline
3	Position measurement uncertainty σ_{z_0} of track 1
4	Position measurement uncertainty σ_{z_0} of track 2
5	Pseudorapidity η of track 1
6	Pseudorapidity η of track 2
7	Number of other tracks crossing the beamline between tracks 1 and 2
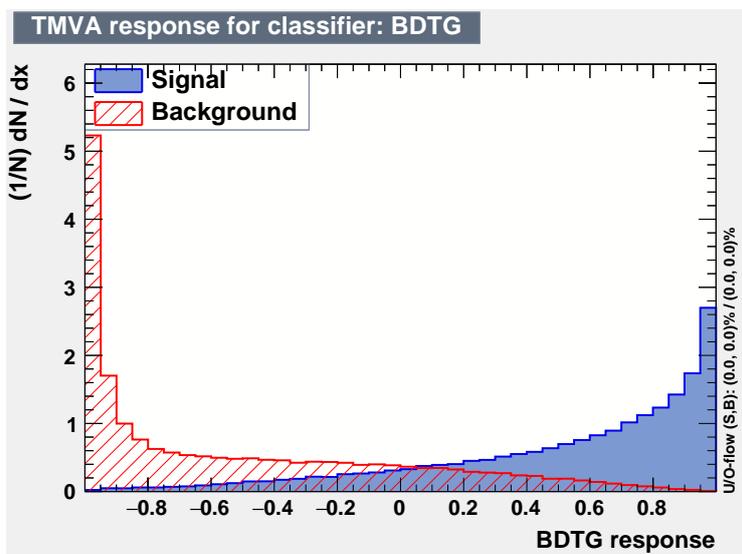
Figure 5. Example BDTG classification weight distributions for true and fake edges.

depends on the range of the track-track distance significance S, see above. Therefore, the value of the intercept β_0 in this approach can be modified in some range to achieve an over- or undersegmentation in order to validate its optimality. This will be further discussed in section 7.3. A one-dimensional regression is tested in this paper, using variable (1) from table 3. The logistic regression for the edge weight calculation is illustrated in figure 6.

The usage of the track-track distance significance for the graph partitioning does not guarantee the compactness — the limited size and the absence of overlap with other clusters — of the obtained track clusters in the Cartesian space, which may be beneficial when the vertex density is large. The compactness requirement can be imposed using the LMC constraint mechanism. Some edges in the connectivity graph can be additionally labelled as "have to be cut", based on a priori information, different from the edge probability itself. To make track clusters more compact, we can constrain the edges to be cut if the corresponding Cartesian track-track distance is larger than some scale. In the following, a rather weak requirement of $|z_i - z_j| < 1$ mm will be used, which removes tracks with very large errors, see figure 2b. In addition to improving the quality of the solution, the constraint limits the phase space of possible solutions, and this leads to a significant algorithm speedup.
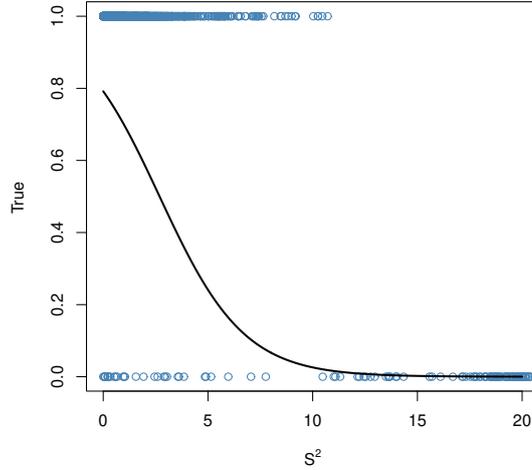
Figure 6. Example one-variable logistic regression for true (top circles) and fake (bottom circles) edges using the squared track-track distance significance S^2.

6 Performance metrics

For a quantitative assessment of the performance of the vertex-finding algorithm, one or several metrics are to be established. To compare the performance of the clustering algorithms in, e.g., image segmentation problems, metrics are usually employed, which are based on the comparison of the obtained assignment of the participating objects to clusters with the truth. One example of such a metric is the Variation of Information (VI) proposed in Reference [23]. The VI metric calculates the degree of compatibility of a clustering C with another clustering C' as

$$VI(C,C') = H(C) + H(C') - 2 \cdot I(C,C') \tag{6.1}$$

with

$$H(C) = -\sum_{k=1}^{K} P(k) \cdot \log(P(k)) \ \text{ and } \ I(C,C') = \sum_{k=1}^{K}\sum_{k'=1}^{K'} P(k,k') \cdot \log\left(\frac{P(k,k')}{P(k)P(k')}\right). \tag{6.2}$$

Here $P(k) = n_k/N$, $P(k,k') = |C_k \cap C'_{k'}|/N$, n_k is the number of nodes in the cluster C_k, N is the total number of nodes in the graph, and K and K' are the number of elements in C and C', respectively. In our case, the VI metric can be used to compare the truth track-to-vertex assignment with the obtained clustering solution. When the obtained set of track clusters and the track-to-cluster assignment reproduce the truth exactly, VI vanishes. Consequently, smaller VI values correspond to more truth-like (and therefore better) clustering solutions.

Another track-to-cluster-based metric, which is investigated in the following, is the Silhouette [24] score

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{6.3}$$

with

$$a(i) = \frac{1}{n_k - 1} \sum_{j,\ i \neq j}^{C_k} d(i,j) \quad \text{and} \quad b(i) = \min_{C_{k'} \neq C_k} \frac{1}{n_{k'}} \sum_{j}^{C_{k'}} d(i,j) \tag{6.4}$$

for node i in cluster C_k. Here $d(i,j)$ is a distance between nodes i and j. In this study, we use the Cartesian distance between tracks and average over all tracks *silhouette* value $\langle s(i) \rangle$ as a quality estimator of the clustering solution. The *silhouette* value is limited $-1 < s(i) < 1$, larger values corresponding to more compact clusters, better separated from each other.

Several other metrics for the assessment of the clustering performance can be found in Reference [23]. These metrics are expected to encounter problems in the present case due to the truth track overlap, as explained in section 4. Tracks are assigned most probably to the wrong cluster by any partitioning algorithm if placed in between tracks from other clusters by mismeasurement. This phenomenon inevitably reduces the accuracy of any track-to-cluster-based metrics. Nevertheless, at least the clustering of the well-measured tracks should reproduce the truth closely, which the track-to-cluster metrics can still be sensitive to.

As the metric accuracy is compromised by the presence of tracks with large measurement errors, it might be useful to downscale the contribution of such tracks to the metric. For the VI metric this can be achieved by weighting every track with σ^{-2} in the metric calculations, namely $n_k \rightarrow n_k = \sum_{i=1}^{k} \frac{1}{\sigma_i^2}$, $N \rightarrow \mathcal{N} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2}$, etc.. For the Silhouette metric the Cartesian distance between two tracks can be replaced by its significance $d(i,j) = S_{ij}$. The weighted versions of the VI and Silhouette metric will be used in the following, along with the original versions.

The number of reconstructed clusters and the weighted average positions of these clusters, dominated by the well-measured tracks, are mostly decoupled from the details of the track-to-cluster assignment. The number of clusters can be directly used as a metric (up to the possible presence of fake clusters), but a Cartesian distance-based metric is not straightforward. One may try to introduce such a metric exploiting the cluster-cluster resolution R_{cc}, i.e. the minimal distance between two reconstructed clusters, see figure 7. The *good, merged, bad* cluster categories could be defined based on whether the cluster-truth vertex distance is smaller or larger than R_{cc}. Such cluster categories could be used to compare various clustering solutions. But this categorisation explicitly depends on R_{cc}, which itself depends on the clustering algorithm. To avoid such circular dependence, a scale-independent Cartesian distance-based metric is needed.

To construct such a metric, we propose the following procedure. Every reconstructable truth vertex is linked to the closest reconstructed cluster in the Cartesian space that has 2 or more assigned tracks. Thus, a list of linked reconstructed clusters is obtained. Then, every reconstructed cluster is classified depending on how many times it enters into this list. If a cluster enters this list only once, there is just a single truth vertex referencing this cluster. Therefore it can be called *unique*, which means that a truth vertex is unambiguously reconstructed as a cluster. If a cluster enters several times into the list, it is referenced by several truth vertices, and therefore it combines tracks from these vertices: this cluster can be called *merged*. Also, some clusters may not appear in this list at all: such clusters are not referenced by any truth vertex and are thus *fake*. The total number of obtained clusters and their classification as *unique, merged, fake* are scale-independent and can be used as a metric to compare various clustering options.
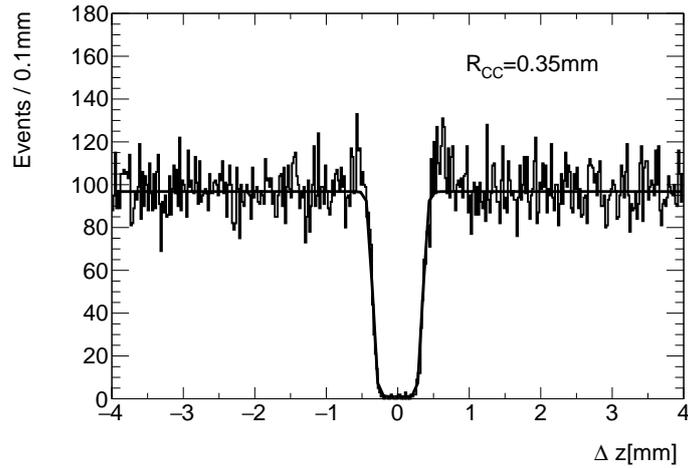
Figure 7. Example of a fit to the cluster-cluster distance to determine the resolution. The used fitting function is $a/\{1 + \exp[b \cdot (R_{cc} - |x|)]\} + c$ where *a, b, c* are free fitting parameters and R_{cc} is the cluster-cluster resolution, defined as the half-width at the half-depth of the dip in the centre of the cluster-cluster weighted centre distances, averaged over all clusters.

7 Results

7.1 LHC Run-2 13 TeV data

First, the LMC clustering algorithm is tested with simulated DELPHES data at a collision energy of 13 TeV, with pileup $\langle\mu\rangle = 63$ and $\sigma_z = 35$ mm. These parameters are chosen to provide simulated data close to the actual data collected by the ATLAS detector in Run 2. Due to the very small transverse width of the ATLAS proton-proton interaction region ($\leq 10\,\mu$m), this width is neglected in the simulation, i.e. $\sigma_{x,y} = 0$. Edge-weight distributions for various edge-labelling approaches on these data are shown in figure 8. The performance of the LMC algorithm on these data is shown in table 4. The rows labelled "cnst" in these tables provide performance estimation with the applied constraints $|z_i - z_j| < 1$ mm, while the "base" rows describe the baseline algorithm performance without constraints.
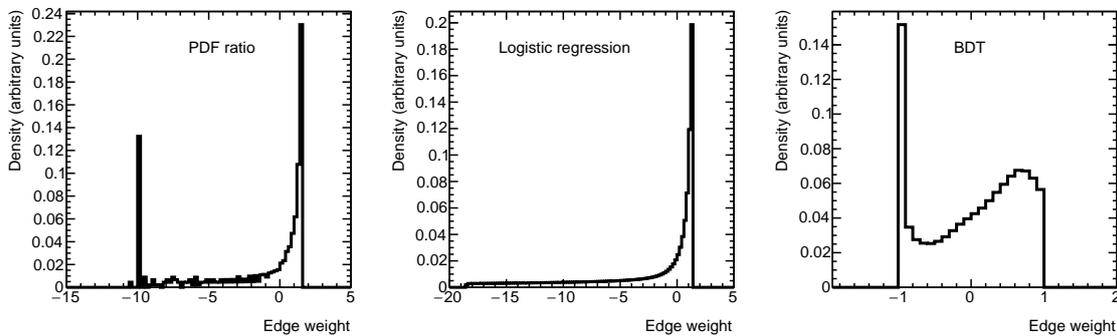
Figure 8. Typical edge weight distributions for various edge labelling options.

As mentioned above, the primary-vertex reconstruction problem is effectively one-dimensional, all tracks and vertices being located along the beam-line. In an ideal case, the reconstructed clusters are strictly separated on this line, i.e. they do not contain tracks assigned to another cluster within their geometrical borders. However, the LMP clustering is based on the distance significance and not on the Cartesian distance itself. Thus, a track may be assigned to another cluster nearby. The column N_{trk}^{wrong} in table 4 shows the fraction of tracks in an event assigned to a cluster, but entirely surrounded by tracks from another cluster along the line. This number is an estimator of the degree of cluster overlap in the obtained solution. The relevant truth track overlap fraction in the same events can be found in table 1 for comparison. The truth track overlap fraction characterises the initial complexity of the event, while the cluster overlap fraction characterises the quality of the obtained solution. In addition, table 9 in the appendix gives the number of isolated nodes (tracks) reported by the LMC clustering algorithm. These non-assigned tracks do not represent the one-track truth vertices, considered non-reconstructable without a priori information, but rather reflect the clustering problems.

Table 4. LMC performance for the collision energy 13 TeV, pileup 63 and interaction region width $\sigma_z = 35$ mm. These simulation parameters are chosen to match the full ATLAS simulation for Run 2 results used for comparison. The column N_{trk}^{wrong} shows the fraction of tracks wrongly associated by the clustering algorithm, which can be compared to the truth track overlap fraction of 20% (table 2).

Edge weight		VI	VI weighted	Silhouette	Silhouette weighted	Unique	Merged	Fake	N_{trk}^{wrong}	CPU
PDF ratio	base	0.839	0.407	0.615	0.646	33.3	8.2	2.4	15%	0.25s
	cnst	0.782	0.362	0.649	0.660	33.9	7.9	2.3	8%	0.18s
Regression	base	0.860	0.416	0.589	0.623	34.7	7.6	4.1	14%	0.27s
	cnst	0.829	0.387	0.614	0.633	35.0	7.5	3.9	8%	0.18s
BDT	base	0.945	0.399	0.478	0.230	35.0	7.5	7.1	5%	0.23s
	cnst	0.937	0.377	0.487	0.234	35.2	7.4	7.0	4%	0.14s

The PDF ratio and the regression-based edge weight assignment result in approximately equal clustering performance. The BDT-based edge weight assignment leads to a significantly worse Silhouette metric value, a smaller value of the cluster overlap and a larger amount of fake clusters. As expected, the weighted versions of the VI and Silhouette metrics have significantly better values than the standard ones due to downscaling of the noise. Using constraints uniformly improves all quality estimators and provides ~ 30% CPU reduction.

In total, 70% of the reconstructable truth vertices are reconstructed as *unique* clusters, while the remaining 30% (i.e. 15) truth vertices are squeezed into 7.5 *merged* vertices. The amount of fake clusters is in the range of 5–15%. The number of tracks in the different cluster categories is presented in figure 9. The number of tracks in the *unique* clusters is close to the track amount in the truth vertices, see figure 2, while the *merged* clusters contain much more tracks. Finally, *fake* clusters have a very small number of tracks.
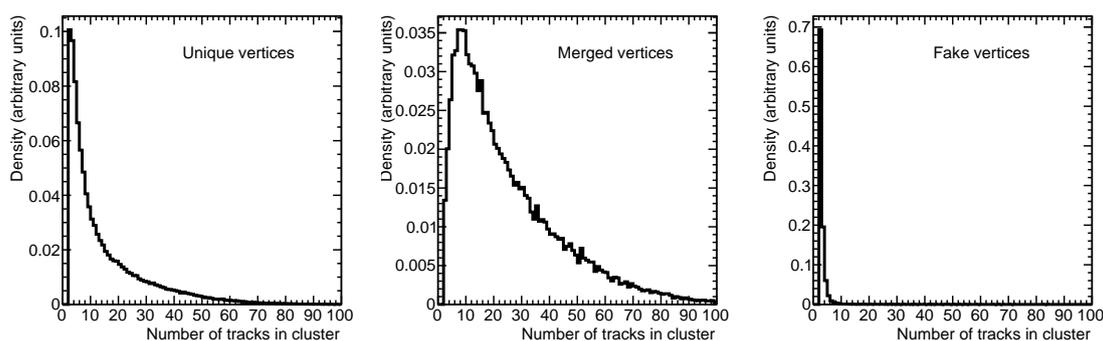
Figure 9. Number of tracks in a cluster for the *unique, merged* and *fake* cluster categories. The distributions are obtained for pileup $\langle\mu\rangle = 63$ data using a one-variable logistic regression for the edge weight assignment.

7.2 High-Luminosity LHC 14 TeV data

The High Luminosity LHC (HL-LHC) project foresees a significant increase in interaction rates to collect significantly more data and thus increase the sensitivity for new physics. The exact parameters of the upgraded HL-LHC are not yet final; pileup values of 150, 200, and 250, and an interaction region width of $\sigma_z = 42$ mm [19] are considered in this paper. These options result in an increase in the density of pileup interaction vertices up to a factor of 4, as compared to the current LHC parameters. The truth track overlap fraction rises from 20% to 66%, see table 1. It is interesting to check the performance of the LMC problem formulation in such extreme conditions.

For this test, the same PDF ratio and logistic regression function are used for the edge weight calculation, while the BDT classification is retrained using $\mu = 150, 200, 250$ data. Results for nominal PDF ratio and logistic regression-based edge weight calculation functions are shown in tables 5, 6, and 7.

Table 5. LMC performance for pileup $\mu = 150$ in an HL-LHC environment with collision energy 14 TeV and interaction region size $\sigma_z = 42$ mm. The column $N_{\text{trk}}^{\text{wrong}}$ shows the fraction of the tracks, wrongly associated by the clustering algorithm, which can be compared to the truth track overlap fraction 41% (table 2).

Edge weight		VI	VI weighted	Silhouette	Silhouette weighted	Unique	Merged	Fake	$N_{\text{trk}}^{\text{wrong}}$	CPU
PDF ratio	base	1.318	0.690	0.535	0.577	57.7	27.4	4.8	28%	1.1s
	cnst	1.211	0.612	0.581	0.609	59.4	26.9	4.1	14%	0.42s
Regression	base	1.316	0.682	0.514	0.559	63.0	25.6	8.8	26%	0.73s
	cnst	1.259	0.634	0.546	0.582	63.6	25.4	8.2	14%	0.50s
BDT	base	1.303	0.658	0.394	0.146	61.8	25.9	13	9%	0.96s
	cnst	1.275	0.616	0.409	0.155	62.8	25.6	12	7%	0.43s

Similarly to the $\mu = 63$ results, the BDT-based edge weight assignment leads to a significantly worse Silhouette metric value, a much smaller value of the cluster overlap and a larger number of fake clusters, while the PDF ratio and regression-based edge weight calculation approaches provide

Table 6. LMC performance for pileup $\mu = 200$ in an HL-LHC environment with collision energy 14 TeV and interaction region size $\sigma_z = 42$ mm. The column $N_{\text{trk}}^{\text{wrong}}$ shows the fraction of the tracks, wrongly associated by the clustering algorithm, which can be compared to the truth track overlap fraction 53% (table 2).

Edge weight		VI	VI weighted	Silhouette	Silhouette weighted	Unique	Merged	Fake	$N_{\text{trk}}^{\text{wrong}}$	CPU
PDF ratio	base	1.574	0.852	0.500	0.546	64.3	40.3	5.7	36%	2.3s
	cnst	1.441	0.756	0.552	0.586	66.6	39.8	4.8	18%	0.69s
Regression	base	1.546	0.825	0.492	0.539	70.3	38.6	9.0	32%	2.4s
	cnst	1.470	0.765	0.529	0.568	71.0	38.4	8.1	18%	0.69s
BDT	base	1.512	0.805	0.312	0.040	69.9	38.6	15.6	13%	1.8s
	cnst	1.479	0.755	0.332	0.051	71.3	38.2	15.0	7%	0.66s

Table 7. LMC performance for pileup $\mu = 250$ in an HL-LHC environment with collision energy 14 TeV and interaction region size $\sigma_z = 42$ mm. The column $N_{\text{trk}}^{\text{wrong}}$ shows the fraction of the tracks, wrongly associated by the clustering algorithm, which can be compared to the truth track overlap fraction 66% (table 2).

| Edge weight		VI	VI weighted	Silhouette	Silhouette weighted	Unique	Merged	Fake	$N_{\text{trk}}^{\text{wrong}}$	CPU
PDF ratio	base	1.782	0.990	0.477	0.526	68.7	53.2	6.4	42%	3.0s
	cnst	1.638	0.887	0.531	0.569	71.0	52.7	5.3	21%	1.7s
Regression	base	1.753	0.961	0.467	0.517	77.1	51.2	11.	38%	3.2s
	cnst	1.672	0.895	0.505	0.547	77.8	51.1	9.9	21%	1.7s
BDT	base	1.691	0.941	0.307	0.040	72.8	52.4	15.	12%	3.0s
	cnst	1.651	0.882	0.330	0.055	74.5	52.0	14.	9%	1.2s

similar performances. The weighted versions of the VI and Silhouette metrics have significantly better values than the standard ones due to downscaling of the noise. The use of constraints significantly improves all quality estimators and provides ∼ 30% CPU reduction.

The number of unambiguously reconstructed *unique* clusters is 53% (44%, 37%) out of the total amount of the reconstructable truth vertices for the pileup $\mu = 150$ (200, 250). The remaining 56 (90, 125) reconstructable truth vertices are clustered into 25 (40, 52) *merged* clusters. The correctness of representation of the initial truth vertices by *merged* clusters is not granted. Truth vertices with a large number of tracks might "absorb" vertices with a small number of tracks.

7.3 LMC performance adjustment

As can be seen from tables 4–7, different edge weight assignment approaches lead to non-coinciding clustering results. For a practical application of the LMC approach for primary vertex finding in the LHC experiments, it is important to verify whether a unique optimal clustering solution exists in this problem and, if so, whether the different LMC cost functions can be tuned to provide the same clustering. As explained in section 5, parameters of the PDF ratio and regression function for the

edge weights can be modified to enforce under- or over-segmentation.The *PDF* ratio function can be scaled up and down. In the logistic regression function, the intercept term can be shifted by a constant. The cost function modifications are tried on the $\mu = 150$ data. The obtained clustering results are shown in figure 10 and figure 11.
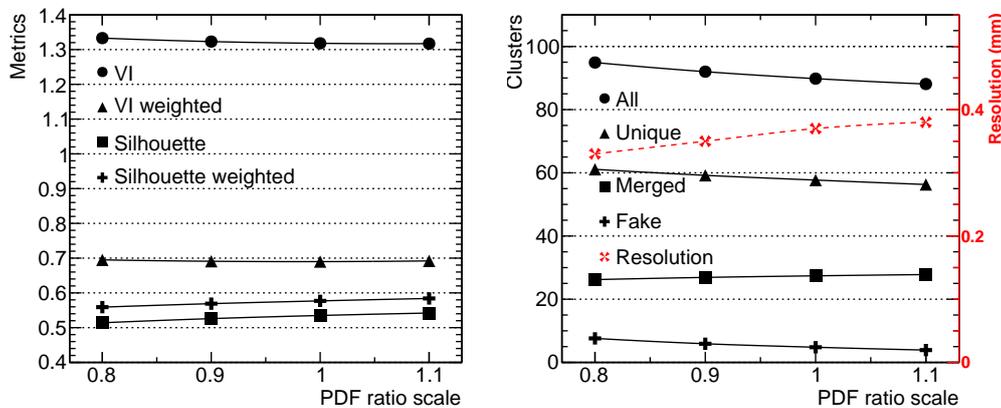
Figure 10. PDF ratio cost-based clustering results as a function of the applied scaling.
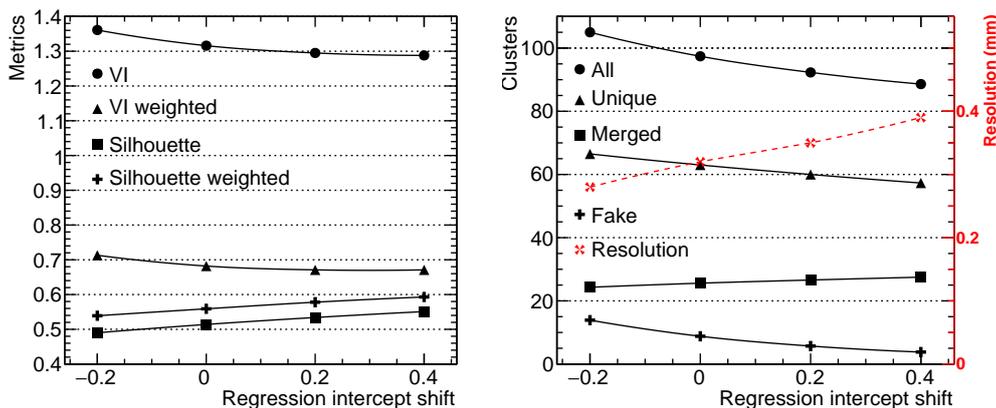
Figure 11. Logistic regression cost-based clustering results as a function of the logistic regression intercept term shift.

In the performed test, the exploited metrics change monotonically depending on the scale factor for the PDF ratio and the intercept shift for the linear regression function. It doesn't seem possible to adjust the PDF ratio and logistic regression parameters so that both approaches provide exactly the same clustering performances in all used metrics. In addition, the BDTG-based Silhouette and Silhouette weighted metrics results (see table 5) are not reproducible by any modification of the PDF ratio and logistic regression cost functions. However, the overall variations of the clustering results remain limited, which means that the LMC approach performance stays close to optimal in the full scanned parameter range.

To conclude, the cost function modification test doesn't demonstrate the presence of an evident unique globally optimal clustering solution for the problem in consideration. Three used edge weight assignment strategies provide different clustering results, which can be additionally changed by simple modification of the cost functions. Therefore, for a practical application as a primary vertex finder, an exact LMC formulation should be chosen based on desired physics requirements, e.g. minimal amount of fake vertices or best vertex-vertex resolution, disregarding the clustering metrics.

7.4   Influence of tracks with large measurement errors

Table 8. Number of selected tracks and the truth track overlap fraction as a function of the track error cut for $\mu = 150$ data.

Track error cut	N_{trk}	Truth track overlap
-	1674	41%
0.8	1540	31%
0.6	1444	27%
0.4	1283	22%

As the initial truth track overlap in an event is caused by the track position mismeasurement, the overlap degree can be reduced by removing the badly measured tracks by cutting on the track measurement error shown in figure 2b. A moderate decrease in the total amount of tracks due to this rejection should not significantly affect the overall clustering efficiency as the total amount of tracks per truth vertex is big enough, see figure 2a. The reduction of the amount of selected tracks and the truth track overlap fraction due to strongly mismeasured track removal is shown in table 8. The results of the clustering are shown in figure 12 for the PDF ratio cost function and in figure 13 for the nominal logistic regression cost function.
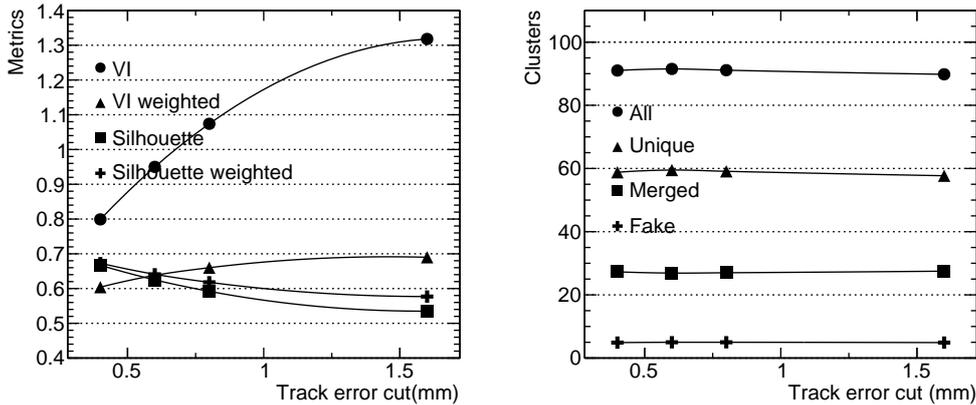
Figure 12. PDF ratio cost-based clustering results as a function of the applied track error cut for the $\mu = 150$ data.

The distance-based metric demonstrates very small changes in the clustering results in a wide range of the badly measured track admixture and, correspondingly, the initial degree of the
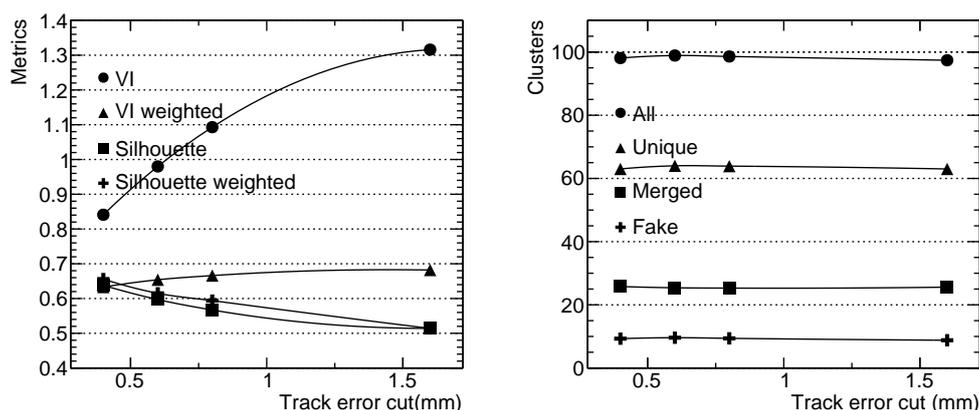
Figure 13. Logistic regression cost-based clustering results as a function of the applied track error cut for the $\mu = 150$ data.

vertex overlap. One may conclude that the amount of clusters identified by the LMC algorithm is largely defined by the tracks with small measurement errors and, therefore, is stable with respect to significant track noise admixture. Redistribution of the tracks with big errors over the obtained clusters doesn't change their amount but evidently strongly affects all track counting-based clustering metrics. The track weighting does mitigate this effect for the VI metric, its weighted version is practically independent of the track noise admixture. Surprisingly, the Silhouette metric is only weakly sensitive to this noise.

7.5 Comparison with the existing approaches

The ATLAS Collaboration used the IVF algorithm [3] to reconstruct the pp collision vertices in Run 1 and the AMVF algorithm [4] in Run 2 and Run 3. Essential characteristics of a primary-vertex reconstruction algorithm are the vertex-vertex resolution and the number of reconstructed vertices as a function of the number of pp interactions. The upper plots in figure 14 present the corresponding distributions for typical ATLAS data for the AMVF and IVF algorithms. The bottom plots show the same distributions provided by the LMC algorithm with the PDF-ratio based edge-weight assignment using DELPHES simulation tuned to the same pileup conditions and track resolutions. The fast DELPHES simulation lacks some features present in the full ATLAS simulation (fakes, inefficiencies, non-Gaussian tails, etc.). However, these features are not expected to change the primary vertex reconstruction results significantly, in particular, due to the highly efficient ATLAS track reconstruction.

Figure 14 clearly demonstrates that the LMC algorithm outperforms the ATLAS heuristic algorithms. It provides significantly better vertex-vertex resolution. This naturally leads to a larger amount of *Unique/Matched* vertices reconstructed by LMC, while the amount of *Merged* vertices remains practically the same. Routine application of the LMC for the primary vertex reconstruction can provide a significant gain in performance for LHC and future collider experiments.
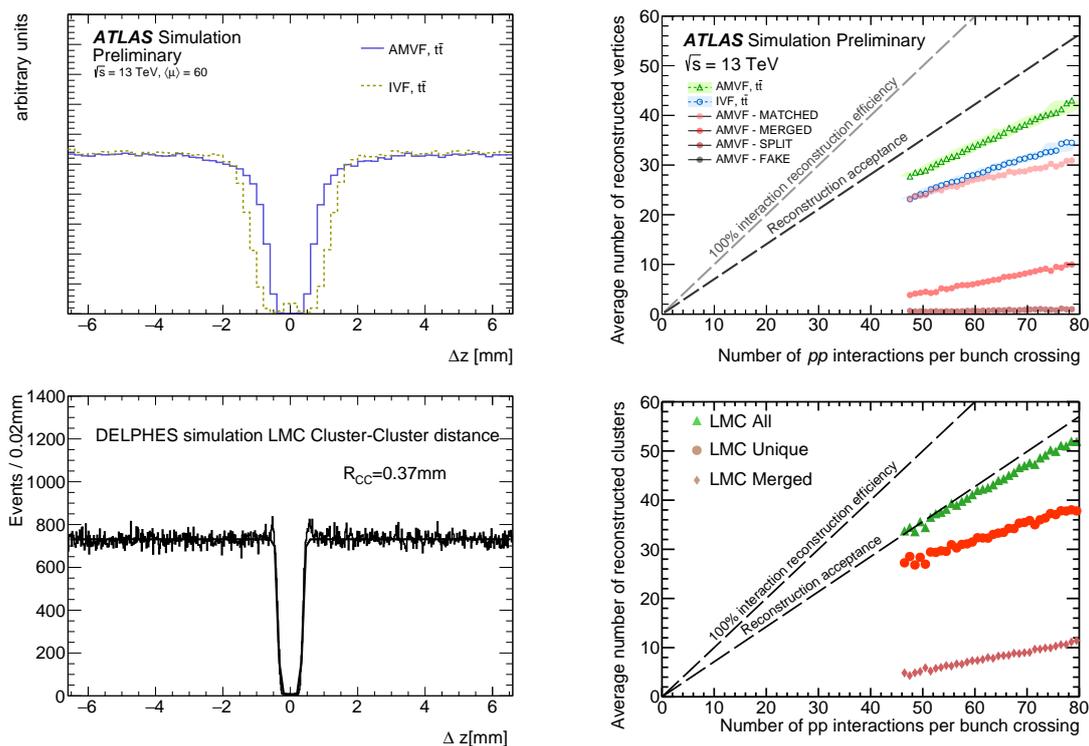
Figure 14. The vertex-vertex resolution and the number of reconstructed vertices as a function of the number of pp interactions for typical ATLAS data. The upper plots are obtained with the ATLAS baseline AMVF [4] and IVF [3] algorithms. The bottom plots are obtained using the LMC algorithm with the PDF ratio-based edge weight assignment on DELPHES $\mu = 63$ data. The DELPHES $\mu = 63$ simulation is specially tuned to match the ATLAS data used in [4]. The cluster-cluster resolution for the LMC algorithm on the bottom left picture is obtained as described in section 6.

8 Conclusion

In this work, we have addressed a typical particle physics problem of reconstructing multiple interaction positions in a dense environment, where each interaction is represented by a cluster of tracks. Significant track reconstruction errors lead to a large overlap of truth track clusters, which makes their identification challenging. Heuristic algorithms are usually used to address this problem. In contrast, we propose to address this problem through a principled formulation as a minimum-cost lifted multicut problem. We construct several cost functions for the LMC from track-track distances and their significance. We study the performance of the LMC algorithm for different vertex densities, cost functions, constraint usage and varying degree of overlap. To address potential performance problems of existing track counting clustering metrics for strongly overlapped clusters, dedicated metrics are introduced.

We demonstrate that the LMC approach outperforms the heuristic algorithms in the problem of vertex reconstruction in dense environments in terms of vertex-vertex resolution and vertex reconstruction efficiency. It works up to the highest vertex density expected at the HL-HLC project in spite of the strong truth track overlap reaching ~ 60%. Variations of the LMC algorithm parameters

and cost functions studied in this work resulted in relatively small variations of the obtained clustering solutions. The developed metrics and the freedom in the choice of the edge-weight assignment strategy allow to fine-tune this algorithm to a specific particle-physics experiment.

Acknowledgments

A   Non-clustered tracks and total reconstructed clusters

In this study, we use four simulated event samples representing realistic proton-proton interactions at the LHC with different energies and luminosities. The total amounts of interaction vertices with one reconstructed track and two and more tracks are shown in table 9. Due to the track measurement errors, the one-track vertices are difficult to reconstruct correctly without a priori information. Finding two and more track vertices becomes problematic if the vertex-vertex distance is less than the typical track measurement error. Both problems are illustrated in table 9, where the amounts of the one-track and multi-track clusters are given for every cost function and event sample.

Table 9. Average numbers of non-clustered tracks and reconstructed clusters obtained by the LMC algorithm with different cost functions as compared to the truth numbers of single-track and multi-track vertices. Results are shown for all collision energies and pileup densities.

	13 TeV		14 TeV					
	$\langle\mu\rangle = 63$		$\langle\mu\rangle = 150$		$\langle\mu\rangle = 200$		$\langle\mu\rangle = 250$	
	$N^{\mathrm{vrt}}_{\mathrm{ntrk}=1}$	$N^{\mathrm{vrt}}_{\mathrm{ntrk}>1}$	$N^{\mathrm{vrt}}_{\mathrm{ntrk}=1}$	$N^{\mathrm{vrt}}_{\mathrm{ntrk}>1}$	$N^{\mathrm{vrt}}_{\mathrm{ntrk}=1}$	$N^{\mathrm{vrt}}_{\mathrm{ntrk}>1}$	$N^{\mathrm{vrt}}_{\mathrm{ntrk}=1}$	$N^{\mathrm{vrt}}_{\mathrm{ntrk}>1}$
Truth	4	50	9	119	12	160	16	199
	$N^{\mathrm{clust}}_{\mathrm{ntrk}=1}$	$N^{\mathrm{clust}}_{\mathrm{ntrk}>1}$	$N^{\mathrm{clust}}_{\mathrm{ntrk}=1}$	$N^{\mathrm{clust}}_{\mathrm{ntrk}>1}$	$N^{\mathrm{clust}}_{\mathrm{ntrk}=1}$	$N^{\mathrm{clust}}_{\mathrm{ntrk}>1}$	$N^{\mathrm{clust}}_{\mathrm{ntrk}=1}$	$N^{\mathrm{clust}}_{\mathrm{ntrk}>1}$
PDF ratio	11	44	19	90	23	110	25	128
Regression	13	46	25	97	27	118	31	139
BDTG	43	50	77	101	102	124	104	140

The number of one-track clusters in each case is significantly larger than the truth amount of one-track interaction vertices, especially in the BDT case. They should be thought of as non-clustered tracks, not as reconstructed one-track vertices. The fraction of multi-track clusters found decreases with the interaction vertex density, as expected.

References

[1] R. Frühwirth and A. Strandlie, *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*, Particle Acceleration and Detection, Springer (2021), 10.1007/978-3-030-65771-0.

[2] ATLAS collaboration, *Primary Vertex Selection in VBF Higgs to Invisibles at $\mu = 200$ with the ATLAS Experiment*, IDTR-2019-004 (2019).

[3] ATLAS collaboration, *Reconstruction of primary vertices at the ATLAS experiment in Run 1 proton–proton collisions at the LHC*, *Eur. Phys. J. C* **77** (2017) 332 [arXiv:1611.10235].

[4] ATLAS collaboration, *Development of ATLAS Primary Vertex Reconstruction for LHC Run 3*, Tech. Rep. ATL-PHYS-PUB-2019-015 (2019).

[5] CMS collaboration, *Description and performance of track and primary-vertex reconstruction with the CMS tracker*, 2014 *JINST* **9** P10009 [arXiv:1405.6569].

[6] S. Hageböck and E. von Toerne, *Medical imaging inspired vertex reconstruction at LHC*, *J. Phys. Conf. Ser.* **396** (2012) 022021.

[7] W. Waltenberger, W. Mitaroff and F. Moser, *RAVE—a detector-independent vertex reconstruction toolkit*, *Nucl. Instrum. Meth. A* **581** (2007) 549.

[8] K. Rose, *Deterministic annealing for clustering, compression, classification, regression, and related optimization problems*, *Proc. IEEE* **86** (1998) 2210.

[9] M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox and B. Andres, *Efficient Decomposition of Image and Mesh Graphs by Lifted Multicuts*, in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 7–13 December 2015, pp. 1751–1759 [arXiv:1505.06973].

[10] E.D. Demaine, D. Emanuel, A. Fiat and N. Immorlica, *Correlation clustering in general weighted graphs*, *Theor. Comp. Sci.* **361** (2006) 172.

[11] S. Chopra and M.R. Rao, *The partition problem*, *Math. Program.* **59** (1993) 87.

[12] A. Horňáková, J.-H. Lange and B. Andres, *Analysis and Optimization of Graph Decompositions by Lifted Multicuts*, in *Proceedings of the International Conference on Machine Learning*, Sydney, NSW, Australia, 6–11 August 2017, pp. 1539–1548 [arXiv:1503.03791].

[13] T. Beier, T. Kroeger, J.H. Kappes, U. Kothe and F.A. Hamprecht, *Cut, glue, & cut: A fast, approximate solver for multicut partitioning*, in *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, U.S.A., 23–28 June 2014, pp. 73–80.

[14] J.H. Kappes, B. Andres, F.A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra et al., *A comparative study of modern inference techniques for structured discrete energy minimization problems* *Int. J. Comput. Vis.* **115** (2015) 155 [arXiv:1404.0533].

[15] T. Beier, B. Andres, U. Köthe and F.A. Hamprecht, *An efficient fusion move algorithm for the minimum cost lifted multicut problem*, in *Proceedings of the 14th European Conference on Computer Vision*, Amsterdam, The Netherlands, 11–14 October 2016, pp. 715–730.

[16] A. Kardoost and M. Keuper, *Solving minimum cost lifted multicut problems by node agglomeration*, in *Proceedings of the 14th Asian Conference on Computer Vision*, Perth, Australia, 2–6 December 2018, pp. 74–89.

[17] J. de Favereau et al., *DELPHES 3, A modular framework for fast simulation of a generic collider experiment*, *JHEP* **02** (2014) 057 [arXiv:1307.6346].

[18] C. Bierlich, S. Chakraborty, N. Desai, L. Gellersen, I. Helenius, P. Ilten et al., *A comprehensive guide to the physics and usage of PYTHIA 8.3*, *SciPost Phys. Codebases* (2022) 8.

[19] L. Medina, R. Tomás, G. Arduini and M. Napsuciale, *Assessment of the performance of High-Luminosity LHC operational scenarios: integrated luminosity and effective pile-up density*, *Can. J. Phys.* **97** (2019) 498.

[20] ATLAS collaboration, *ATLAS Insertable B-Layer Technical Design Report*, Tech. Rep. CERN-LHCC-2010-013 (9, 2010).

[21] ATLAS collaboration, *Impact Parameter Resolution Using 2016 MB Data*, IDTR-2016-018 (2016).

[22] A. Höcker et al., *TMVA - Toolkit for Multivariate Data Analysis*, Tech. Rep. CERN-OPEN-2007-007 (2007).

[23] M. Meilă, *Comparing clusterings—an information based distance*, *J. Multivar. Anal.* **98** (2007) 873.

[24] P.J. Rousseeuw, *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis*, *J. Comput. Appl. Math.* **20** (1987) 53.