MidPoint Algorithm for Jet Reconstruction in ATLAS Experiment

A. Cheplakov, A. S. Thompson

Department of Physics and Astronomy
University of Glasgow
Glasgow, G12 8QQ, UK

February 20, 2007

Abstract

The MidPoint algorithm for jet reconstruction was initially developed for Tevatron experiments (CDF and D$\emptyset$). It is now available in ATLAS. To check its performance we have applied the algorithm for jet reconstruction in di-jet events (CSC samples) and $t\bar{t}b\bar{b}$ events. Some important parameters of the algorithm have been optimized. The results of the jet reconstruction using three different algorithms - fast k$_T$, "standard" fixed cone and MidPoint - have been compared.

1 Introduction

Traditionally hadron-hadron experiments have used Cone algorithms to identify jets where a cone jet consists of all particles in a cone size R_{cone} defined in $\eta \times \phi$ space [1]. It is possible to search for all stable cones but to save computation time the search is initialised using seeds formed by the most energetic particles in an event.

Cone algorithms, however, experience problems in particular with infrared and collinear sensitivity. In the first case the presence of soft radiation between jets can lead to the jets being merged as illustrated in Fig. 1a. As an example of collinear sensitivity the need for a seed energy threshold can mean that a jet whose energy is distributed among several calorimeter cells will not produce a seed, as indicated in Fig. 1b. To address the problem of infrared sensitivity seeds were introduced at the vector mid points between jets effectively replacing the soft parton in Fig. 1a with a seed.

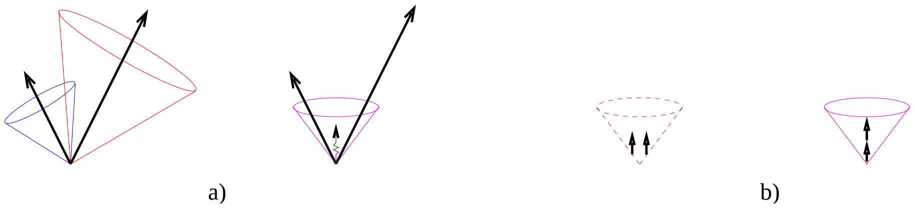

Figure 1: *An illustration of (a) infrared and (b) collinear sensitivity in cone jet clustering (see also paper [1]).*

The MidPoint algorithm is a cone based algorithm developed to cure the infrared instability of the fixed cone approach. It starts by clustering a set of stable cone protojets based on seeds with a reduced cone size, $r \times R_{cone}$, where r is a 'free' parameter ($r < 1$). Extra seeds are then generated at the mid points between pairs of neighbouring protojets if the jet axis directions are separated by a distance, less than $s \times R_{cone}$, where s is another free parameter. Further protojets are then formed with no reduction factor applied to R_{cone}. It should be noted that only pairs of jets were used in generating these mid point seeds. Finally a split-merge algorithm is applied, the protojets are merged if the overlapping p_T of neighbouring jets is greater than a factor m of the smaller p_T (m is yet another free parameter, $m < 1$), otherwise the jets are split with shared cells being assigned to the nearest jet.

The most important parameters used in the MidPoint algorithm are:

- the cone size, R_{cone}, as used in the standard fixed cone algorithm;
- the reduction factor, r, where the initial seed jets have a cone size $r \times R_{cone}$;
- the separation factor, s, where the maximal separation between initial jets to create mid point seed is limited to $s \times R_{cone}$;
- the split-merge factor, m, to determine the p_T fraction on the overlap between jets when they should be merged;
- the minimum p_T for a seed (was set to 2.0 GeV/c).

The initial values for the MidPoint parameters in our analysis were set to be $r = 0.5$, $s = 2.0$, as used in the CDF experiment [3], and $m = 0.5$ as suggested in Ref. [1].

The MidPoint algorithm has a similar performance as the fixed cone with additional potential advantages (infrared stability). Note that two jet finding algorithms, k_T and MidPoint, were recommended for the analysis of Run II data in CDF and DØ experiments.

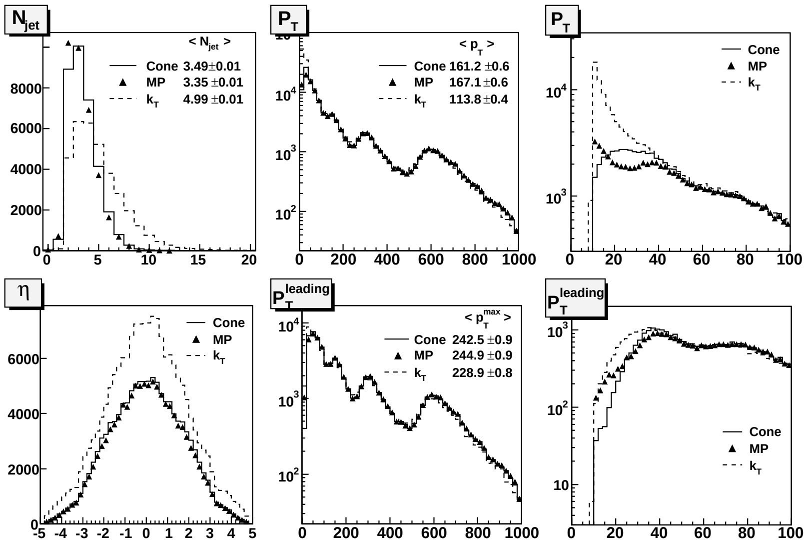

Figure 2: *Comparison of the fixed cone, Midpoint and* k_T *algorithms applied to di-jet events showing jet multiplicity, pseudo-rapidity and transverse momentum for all and leading jet over both the full momentum range and at the low end of the spectrum*

The purpose of the Note is to present a study of the MidPoint algorithm for jet reconstruction in Athena in ATLAS. We also wanted to check its performance in various environments. The results of the study has been compared with the results of other algorithms already used in ATLAS experiment. In Section 2 we have analysed the CSC sets of di-jet events having relatively low jet multiplicity, whereas in section 3 the results are presented for more complex environment of $t\bar{t}b\bar{b}$ events. This set of events we used for optimization of the MidPoint's main parameters. Finally, the "optimized" algorithm was applied for b-tagging study presented in section 4.

2 Di-jet events

The performance of the MidPoint algorithm was studied first by comparing it to that obtained using the Cone and k$_T$ algorithms when applied to di-jet events. About 34K events from the CSC di-jet sets J2-J6 processed using the software release 11.0.41 were passed through Jet Finding including the MidPoint algorithm, which is not a default option, along with the standard Cone and fast k$_T$ [2] algorithms. The default cone size of $R_{cone} = 0.7$ was used, the parameters for the MidPoint were set as above and for the fast k$_T$ algorithm a value R=0.6 was used as recommended [4]. The cell calibration set H1WeightToolRomeHack was applied to all algorithms though this is optimised to the standard cone algorithm at $R_{cone} = 0.7$.

Our results for di-jet events are presented; no weighting was applied when events from various CSC sets were put together. The jet multiplicities, rapidities and p$_T$ are shown in Fig. 2 along with the p$_T$ of the two leading jets, the p$_T$ distributions are additionally shown at low p$_T$ to highlight the differences at the low end of the spectrum. A further constraint was applied for selection of the leading jets such that $\pi/3 < \Delta\phi_{jet-jet} < 2\pi/3$. Fig. 3 shows E-ratio plots, the ratios of the energy of the leading jets found using reconstructed events to the "true" value obtained from Monte-Carlo data at the hadron level as a function of $\mid \eta \mid$ and E^{true}_{jet} and the jet energy resolutions as a function of E^{true}_{jet}. The "true" and reconstructed jets are required to be matched within ΔR=0.1 [1].

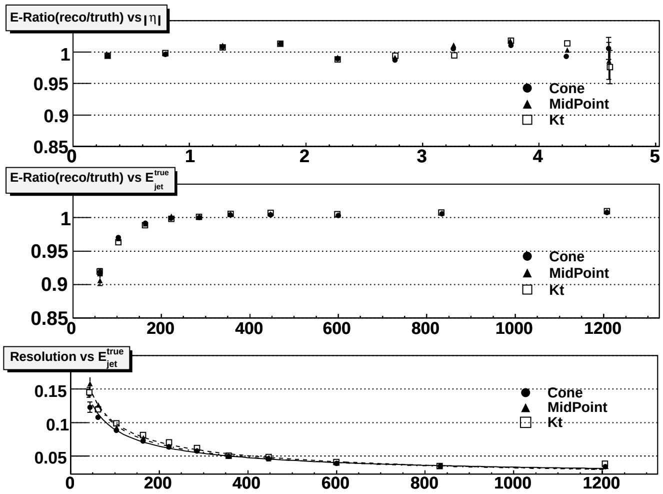

Figure 3: *The relative performance to the jet algorithms as indicated by the ratios of the reconstructed leading jets energy to the "true" value as a function of jet rapidity (top plot) and "true" energy (middle plot), the bottom plot shows the jet energy resolution.*

The k$_T$ algorithm finds more low energy jets than the cone based algorithms but differences between the three algorithms are small when applied to the leading jets.

---

[1] $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is a distance in $\eta \times \phi$ space.

3 Complex environment ($t\bar{t}b\bar{b}$ events)

Once the MidPoint algorithm had been successfully tested on di-jet samples with relatively high jet momenta, it was used for reconstruction of lower energy jets in the complex environment of $t\bar{t}b\bar{b}$ events. These samples are different in terms of jet energies and multiplicities. About 20,000 $t\bar{t}b\bar{b}$ events have been fully simulated and reconstructed with Athena version 11.0.41. Up to 8 jets could be reconstructed in each event if both W's from t-quarks decay hadronically.

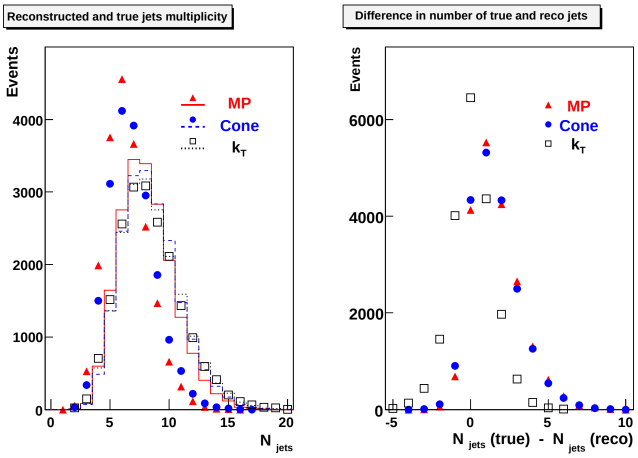

Figure 4: *On the left: multiplicity of the "true" jets reconstructed from Monte-Carlo data (histograms) and "reco" jets from the calorimeter cells by means of Cone, Midpoint and k_T algorithms. On the right: the difference in the number of "reco" jets versus "true" jets.*

All three algorithms - MidPoint, Cone (with $R_{cone}= 0.4$) and fast k_T (R=0.3) - were used for jet reconstruction using the same H1WeightToolRomeHack cell calibration set. A new set of parameters for the MidPoint algorithm was selected after optimization described below in this chapter. In Fig. 4 (left) we have compared the multiplicity of "true" jets reconstructed from the Monte-Carlo data (histogram) and "reco" jets from the calorimeter cells using a seed p_T-threshold of 2 GeV/c. All three algorithms give similar jet multiplicities when applied at the hadron level. But at the "reco" jet level, while the k_T algorithm gives comparable results to "true" jets, the MidPoint and Cone algorithms produce fewer jets (initially by ~1.5 in average) as shown in Fig. 4 (right). The discrepancy between the algorithms becomes less significant when the p_T-threshold is increased, for a p_T cut of 20 GeV/c all three have a very similar deficit in "reco" jets.

The jet finding efficiency (defined as the ratio of the p_T spectra of "true" jets matched to a "reco" jet to the p_T spectra of all "true" jets) as a function of jet p_T for all three algorithms is presented in Fig. 5 for the di-jet sample ($R_{cone} = 0.7$) and for $t\bar{t}b\bar{b}$ events ($R_{cone} = 0.4$). The $t\bar{t}b\bar{b}$ sample results for k_T and cone-based algorithms are in better agreement than the di-jet events which could be due to smaller cone size used for jet reconstruction.

In Fig. 6 (left part) the jet multiplicity, spectra of reconstructed jet p_T, jet pseudorapidity and transverse momenta of the leading jet are shown. In general, k_T and Cone04 both tend to make more jets of low energies, while the difference between the various approaches vanishes for jets with higher energy.

Jet energy values reconstructed by different algorithms were compared to their "true" Monte Carlo values matched via the EventView package [5] ($\Delta R_{match} <0.1$). In Fig. 6 (right part) for all reconstructed jets the ratios of the jet energy to the Monte Carlo "truth" (E-ratio's) are shown as a function of $\mid \eta \mid$ and E_{jet}^{true} along with the jet energy resolution. A spike in E-ratio at low jet energies is due to a 10 GeV threshold for selection of the reconstructed jets (no threshold was applied for

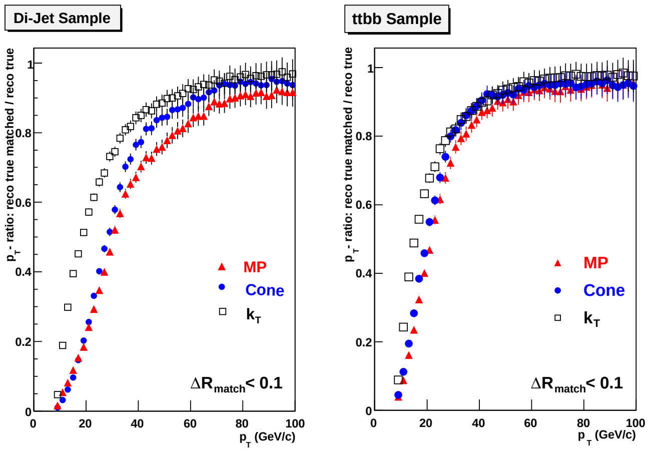

Figure 5: *Jet finding efficiency: the ratio of p_T-spectra of the "true" jets matched to a "reco" jet to p_T-spectra of all "true" jets; on the left - results for di-jet sample and on the right - results for $t\bar{t}b\bar{b}$ events. The $\Delta_{match}R <$ 0.1 cut was applied for jet matching.*

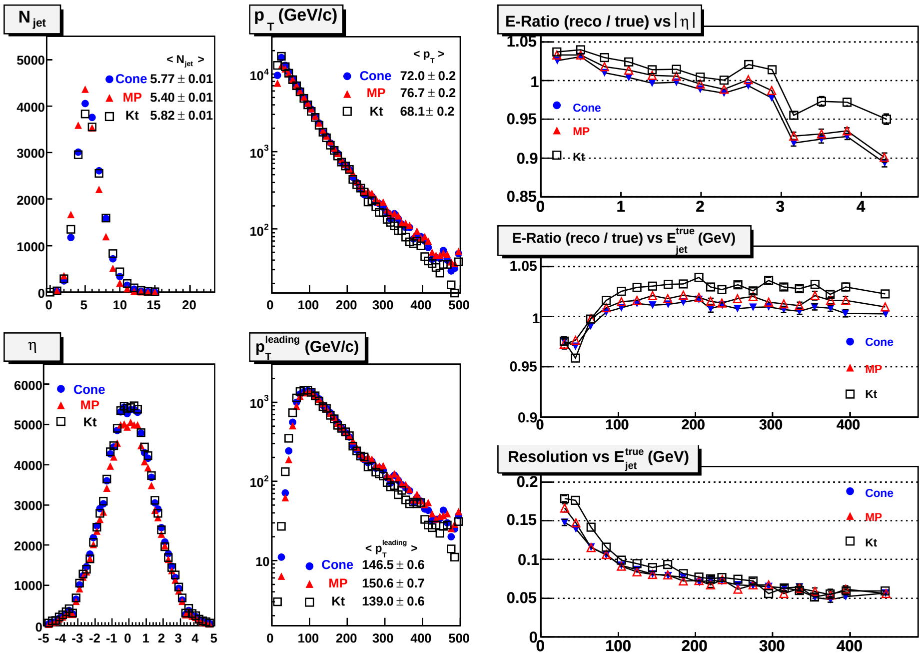

Figure 6: *Comparison of the Cone, Midpoint and k_T algorithms. On the left: jet multiplicity, transverse momentum, rapidity and leading jet transverse momentum spectra in $t\bar{t}b\bar{b}$ events. On the right: ratio of the reconstructed jet energy to the "true" value as a function of jet rapidity (top plot) and its "true" energy (middle plot), bottom plot shows the jet energy resolution.*

the "true" jets). Cone and MidPoint algorithms give similar results for the energy resolution. E-ratio for Midpoint is close to unity within 1÷2%. The observed disagreement between different algorithms could be due to the cell calibration set used (H1WeightToolRomeHack) which was not optimal in each case.

3.1 Optimal values of relevant parameters

The main parameters used in the MidPoint algorithm were altered an attempt to achieve a better agreement for reconstructed jets with the Monte-Carlo data. The initial cone reduction factor r was varied within (0.3÷1.0) interval for s fixed at 2.0, and the seed jet separation factor s was varied within (1.0÷2.4) for r=0.5. Results of the jet reconstruction for maximal and minimal values of the two parameters taken at the edges of the intervals in $t\bar{t}b\bar{b}$ events are presented in Fig. 7. The jet multiplicity varies by 10% due to the changes in r, principally affecting the low energy jets (p$_T$ < 50 GeV/c). It does not influence the p$_T$ of leading jets. The ratio of the reconstructed jet energy to the "true" jet energy varies with r and s for different jet rapidities and energies within 2÷3%. The effect is much smaller for higher p$_T$'s and negligible for the jet energy resolution. From the energy ratio plots one may conclude that lower values of r and higher values of s parameter are preferable since they result in an E-ratio closer to unity.

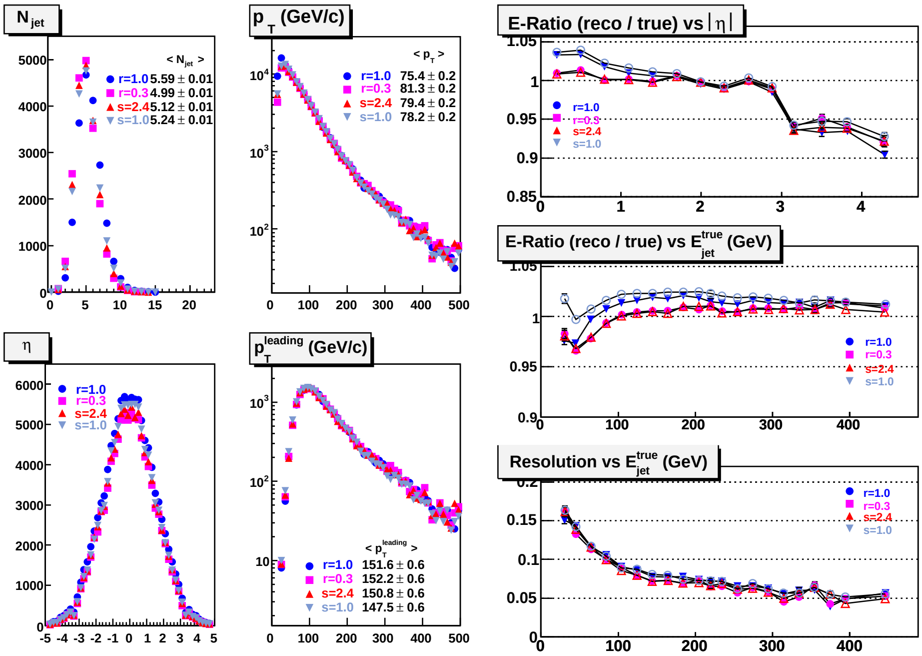

Figure 7: *Summary plots for optimization of r and s parameters of MidPoint algorithm in $t\bar{t}b\bar{b}$ events. On the left: jet multiplicity, transverse momentum, rapidity and leading jet transverse momentum spectra. On the right: ratio of the reconstructed jet energy to the "true" value as a function of (upper plot) jet rapidity and (middle plot) its "true" energy, and jet energy resolution (lower plot). Jets were reconstructed using various r and s values.*

For further optimization the masses of the W boson and Top quark were reconstructed in the $t\bar{t}b\bar{b}$ events using the tools provided in the TopRec package [6]. Events for the analysis were selected according to the following criteria: N$_{jet}$ ≥ 1, N$_{b-jet}$ ≥ 2 and $\not\!\!E_T$ > 20 GeV. An example of the invariant mass spectra for hadronic decays of W and Top is presented in Fig. 8. Jets were reconstructed by means of the MidPoint 0.4 algorithm. Upper plots show the invariant masses for W and Top candidates from the selected event sample. The lower plots show the W and Top masses for the candidates matching the Monte Carlo "true" objects. The spectra were fitted by Gaussian within ±1σ around the peak position and values of the W and Top masses were used as the benchmark for further optimization of the MidPoint algorithm parameters.

First we selected the optimal values of r and s parameters used at the initial stage of jet finding. In Fig. 9 the left two columns show the fitted values of the W and Top masses as a function of r

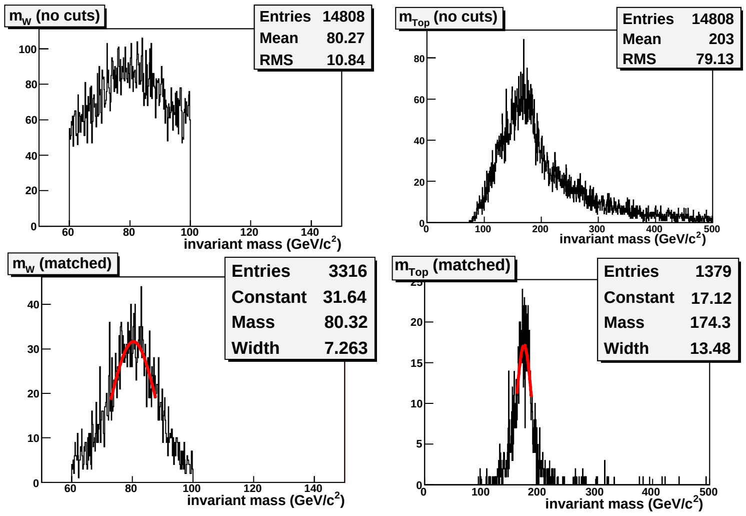

Figure 8: *Invariant masses of W and Top candidates (all hadronic decays, jets are reconstructed by MidPoint algorithm with R_{cone} = 0.4 and main parameters set to r = 0.75, s = 1.6, m = 0.5).*

(in this case s was set to 2.0) and s (r set to 0.5). Based on these results one would prefer higher values of r and smaller values of s to give a better estimate for m$_W$ and m$_{Top}$. Having in mind the preferences revealed in E-ratio plots, we selected the "optimal" values of r=0.75 and s=1.6. Once these parameters were fixed then the masses of W and Top were obtained as a function of the split/merge factor m. Results are presented in the right column in Fig. 9. The value of m = 0.5 looks preferable. Also it is seen that the widths of the W and Top peaks are not sensitive to the changes of the main parameters (see three bottom plots).

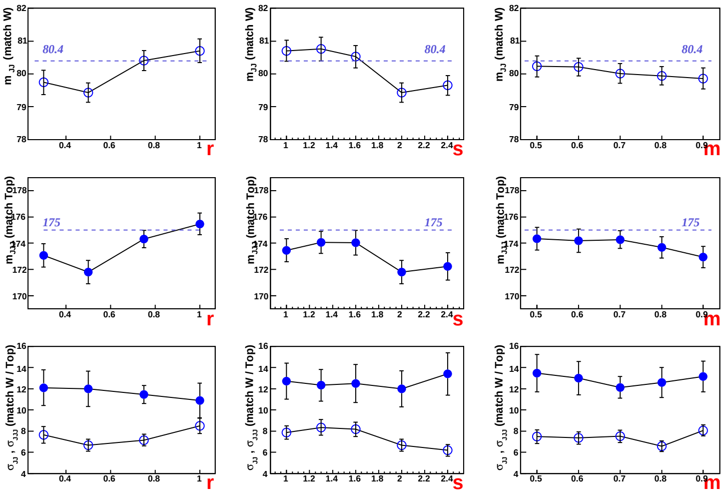

Figure 9: *Invariant masses and widths (bottom plots) of W and Top candidates reconstructed by MidPoint algorithm as a function of r, s and m parameters.*

The jets in the sample of $t\bar{t}b\bar{b}$ events were reconstructed with the "optimized" MidPoint algorithm using selected r, s and m values. The results have been already shown in Fig. 6 together with results for Cone and k$_T$ algorithms. In Table 1 the fit results for 2- and 3-jets invariant mass spectra for all three algorithms are presented. It is seen that the "optimized" MidPoint algorithm provides better agreement with the expected values of m_W and m_{Top}.

Table 1: *W (80.3 GeV) and Top (175 GeV) masses and widths as reconstructed via different algorithms: fixed cone and MidPoint (both with R_{cone} = 0.4) and k_T (with R = 0.3).*

Algorithm	m_W	σ_W	m_{Top}	σ_{Top}
Cone 0.4	79.1±0.3	7.3±0.6	170.2±0.8	11.2±1.6
k_T 0.3	78.0±0.4	8.2±0.7	168.2±1.1	14.6±2.0
MidPoint 0.4	80.3±0.3	7.4±0.6	174.6±0.7	11.9±1.0

4 B-tagging performance

In this chapter we compare b-tagging performance of Cone, MidPoint and k_T algorithms. The same sample of $t\bar{t}b\bar{b}$ events has been used for this study.

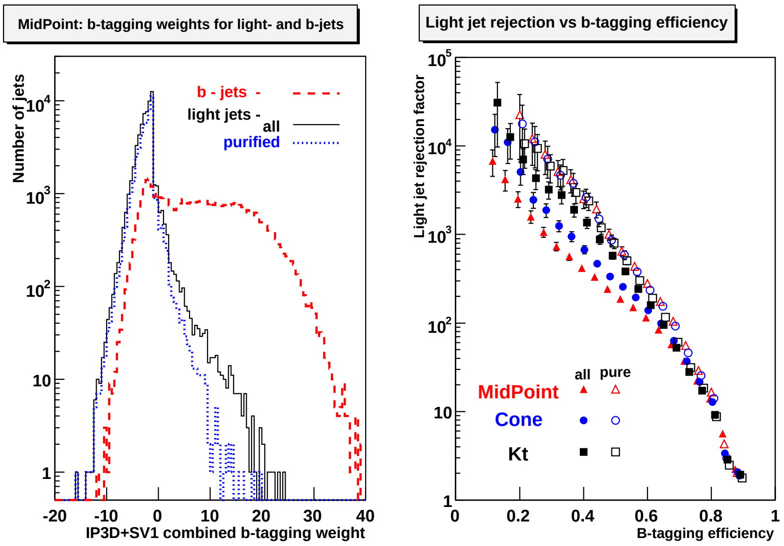

Figure 10: *On the left: combined b-tagging weight for b-jets (dashed line) and two samples of light jets - initial sample (solid line) and "purified" one (dashed line, see text). On the right: light jet rejection factor as a functon of b-tagging efficiency for the initial (solid symbols) and "purified" (open symbols) samples and for all three algorithms.*

To calculate b-tagging efficiency we have used the b-tagging weight obtained for combination of impact parameter-based tagger (IP3D) and secondary vertex-based tagger (SV1) provided by the JetTagBuilder algorithm in Athena. In Fig. 10(left) the combined weight is presented for b-tagged jets and for two samples of light jets - all light jets and "purified" sample of light jets having the distance ΔR >0.8 from b- or c-partons. The distributions are shown for the jets reconstructed by MidPoint algorithm, Cone and k_T show similar results.

In Fig. 10(right) the light jet rejection factor is presented as a function of b-tagging efficiency for the initial sample of light jets and for the "purified" sample, and for all three algorithms. MidPoint gives lower rejection efficiency for the initial sample of light jets, but is slightly better than Cone and k_T for the "purified" sample. Better b-tagging calibration is needed to clarify the algorithm sensitivity to the purification procedure.

5 Conclusions

The performance of the MidPoint algorithm has been studied and compared to the standard Cone and fast k_T algorithms. Different sets of events were used for this study - CSC di-jets with relatively high jet p_T and $t\bar{t}b\bar{b}$ events with much lower jet energy and higher jet multiplicity.

It has been shown that the reconstruction efficiency of MidPoint is as good as that of the Cone algorithm and is similar to k_T, though Cone and k_T tend to make more low energy jets contributing to combinatorial background.

B-tagging performance of all three algorithms is similar.

Optimization of the MidPoint's relevant parameters influenced jets with energy $E_{jet} < 50$ GeV and provided good agreement with the Monte-Carlo "true" values for jet energies and expected masses of W-boson and Top-quark. More accurate optimization of the MidPoint algorithm and detail comparison with other jet reconstruction algorithms in ATLAS are expected when better jet calibration tools become available.

References

[1] G.Blazey et al., "Run II Jet Physics", hep-ex/0005012 v2 (2000);

[2] M.Cacciari and G.P.Salam,"Dispelling the N^3 myth for the k_T jet finder", hep-ph/0512210 v2 (2006);

[3] D.Acosta et al., "Study of JetShapes in Inclusive Jet Production in $p\bar{p}$ Collisions as $\sqrt{s} = 1.96$ TeV", ep-ex/0505013 v2 (2005);

[4] P.A. Delsart, "Effect of varying parameters for Kt jets", slides shown at JetRec Phone Conference (19 April 2006): $\underline{http://indico.cern.ch/conferenceDisplay.py?confId=a062118}$;

[5] see EventView TWiki at $\underline{https://uimon.cern.ch/twiki/bin/viewauth/Atlas/EventView}$, many thanks to A.Farbin for kind assistance;

[6] see TopView TWiki at $\underline{https://uimon.cern.ch/twiki/bin/view/Atlas/TopView}$;

A Appendix. How to use MidPoint algorithm for jet reconstruction in Athena

The relevant job option files need to be added to the file `RecExCommon/CombinedRec_config.py` to use the MidPoint tool. Its implementation in `JetRec/MidPointTowerJet_jobOptions.py` is shown below including modifying the jet finding parameters:

```
theApp.topAlg += [ "JetAlgorithm/MidPointJets" ]
# -- input container
MidPointJets = Algorithm( "MidPointJets" )
MidPointJets.JetCollectionName = "MidPointTowerJets"
```

```
# -- setup without jet monitoring
MidPointJets.AlgTools = [
    "JetTowerNoiseTool/DoNoise",
    "JetSignalSelectorTool/InitialEtCut",
    "JetMidPointTool/MidPoint",
    "JetCellCalibratorTool/CellCalibrator",
    "JetSignalSelectorTool/FinalEtCut" ]
# -- JetMidPointFinderTool (defaults)
MidPointJets.MidPoint.ConeR        = 0.7
MidPointJets.MidPoint.ReduceCone   = 0.5
MidPointJets.MidPoint.Separation   = 2.0
MidPointJets.MidPoint.MergeCut     = 0.5
MidPointJets.MidPoint.SeedPt       = 2.0*GeV
# -- JetCellCalibratorTool (defaults)
MidPointJets.CellCalibrator.CellCalibratorName = "H1WeightToolRomeHack"
# -- Initial and Final signal selections
MidPointJets.InitialEtCut.UseTransverseEnergy = True
MidPointJets.InitialEtCut.MinimumSignal       = 0*MeV
MidPointJets.FinalEtCut.UseTransverseEnergy = True
MidPointJets.FinalEtCut.MinimumSignal       = 10.*GeV
```

However the tool is included in the GenericJets algorithm and this provides a simple method to implement many instances of the jet finding tools, necessary for tuning jet parameters. An example of using this tool in `RecExCommon/CombinedRec_config.py` is shown here:

```
if doJetRec  and DetFlags.haveRIO.Calo_on() :
    # make the tower beforehand if not already done
    if not doCaloCluster:
        include( "CaloRec/CaloCombinedTower_jobOptions.py" )
    # Tower ProtoJet Reconstruction
    JetAlgs = {
        "KtTowerJets" : [0.45],
        "Kt3TowerJets" : [0.3],
        "Kt6TowerJets" : [0.6],
        "ConeTowerJets": [0.7],
        "Cone4TowerJets": [0.4],
        "MidPointTowerJets": [0.7, 0.5, 2.0],
        "MidPointR75S16M5TowerJets": [0.7, 0.75, 1.6, 0.5]
#                         [coneR,reduceCone,separation,mergeCut]
        ........
        }
    cellCalibrator = "H1WeightToolRomeHack"
    include( "JetRec/GenericJets_jobOptions.py" ).
```