CDF Run 2 Jet Algorithms

Jay R. Dittmann
Baylor University / CDF

TeV4LHC Workshop
QCD Session

December 1, 2004

Outline

① Jets and Jet Algorithms

② Review of the CDF Run I Cone Algorithm

③ Jet Algorithms for Run II

- Midpoint Cone Algorithm

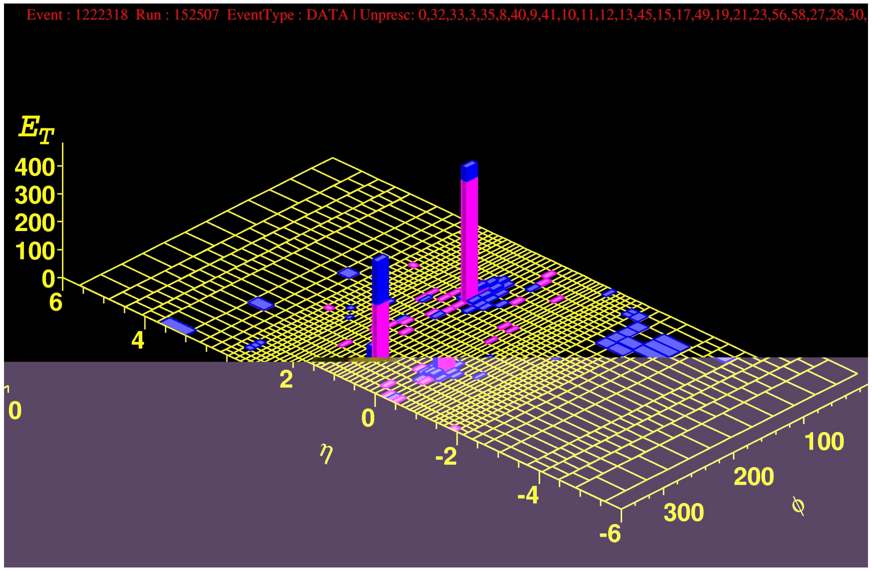
  Fix #1: Search Cone Radius

  Fix #2: Split/Merge Fraction

- K_T Clustering Algorithm

④ Jet Algorithms at CDF

⑤ Summary



Jets and Jet Algorithms

Definition of a Jet

jet (jet) *n.* a collimated spray of high energy hadrons

For *quantitative* studies, a precise definition of a jet is necessary. A precise definition depends on:

- What we think a "jet" is (Theory)
- What we can accurately measure (Experiment)

This leads us to a *Jet Algorithm*.

Two main types of jet algorithms have been used/suggested:

① Cone algorithms

② K_T algorithm

For Run 1, CDF and DØ used cone algorithms almost exclusively for published analyses.

Review of the CDF Run I Cone Algorithm

JETCLU

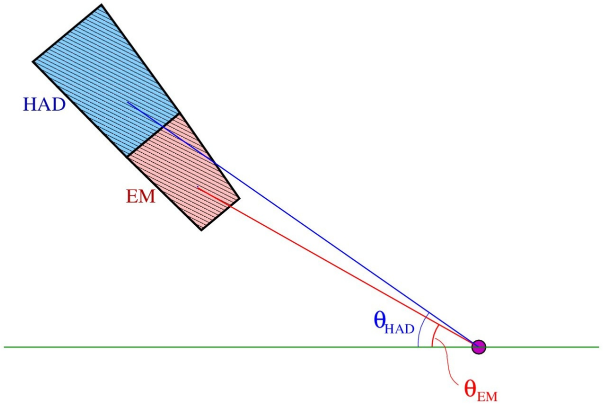
JETCLU is the name of the cone jet algorithm that was used by CDF in Run I. It was comprised of many (rather opaque) FORTRAN subroutines. For Run II, the original Run I FORTRAN code was rewritten in C++ to fit within the new offline code framework.

The JETCLU Algorithm

① Create a list of calorimeter towers with $E_T >$ 1.0 GeV, where

$$E_T \equiv E_{\text{EM}} \sin\theta_{\text{EM}} + E_{\text{HAD}} \sin\theta_{\text{HAD}}$$

These towers are called *seed towers*.



E_{EM} and E_{HAD} denote the EM and HAD energy measured in the respective calorimeter cells. The angles $\sin\theta_{\text{EM}}$ and $\sin\theta_{\text{HAD}}$ specify the location of the EM and HAD cells with respect to a vertex position in z.

Review of the CDF Run I Cone Algorithm

The JETCLU Algorithm (cont.)

② Starting with the highest-E_T seed tower, form *preclusters* by clumping together adjacent seed towers within a cone of radius R in η-ϕ space.

- Every seed tower is assigned to exactly one precluster.
- Each precluster contains at least one seed tower.

③ For each precluster, find the E_T-weighted centroid and draw a cone of radius R around it.

A *cluster* consists of all towers within the cone that have $E_T > 100$ MeV. Iteratively recalculate the cluster centroid, draw a new cone, and add towers until the tower list remains unchanged or the number of iterations reaches a maximum number.

Once a tower is part of a cluster, it is never removed, even if the center of the cone drifts far away. This is called *ratcheting*.

With ratcheting, a cluster may contain towers that are no longer within the cone!

Review of the CDF Run I Cone Algorithm

The JETCLU Algorithm (cont.)

④ Treat overlapping clusters:

- If the towers of one cluster are completely contained within another, the smaller (lesser E_T) of the two clusters is dropped.
- If the tower lists of different clusters partially overlap, an *overlap fraction* is computed by summing the E_T of the common towers and dividing the total by the E_T of the smaller cluster.
- If the overlap fraction is above a cutoff (> 0.75), the two clusters are combined. If the fraction is less than the cutoff, the clusters remain separate, and each tower in the overlap region is assigned to the cluster with the nearest centroid in η-ϕ space.

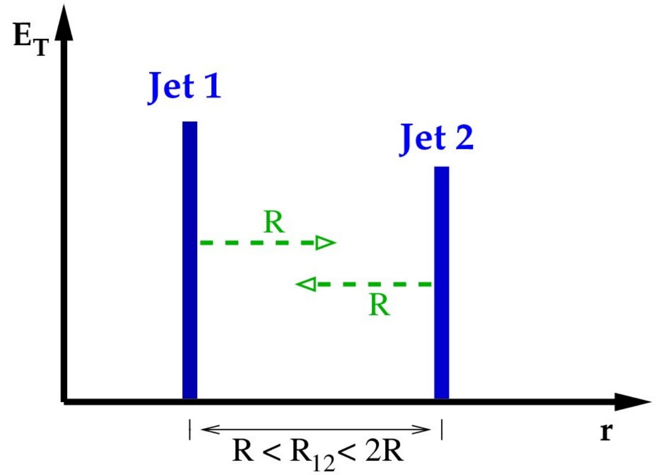
⑤ The final, remaining clusters are *jets*.

At this point, jets are defined by their *tower lists*.

Review of the CDF Run I Cone Algorithm

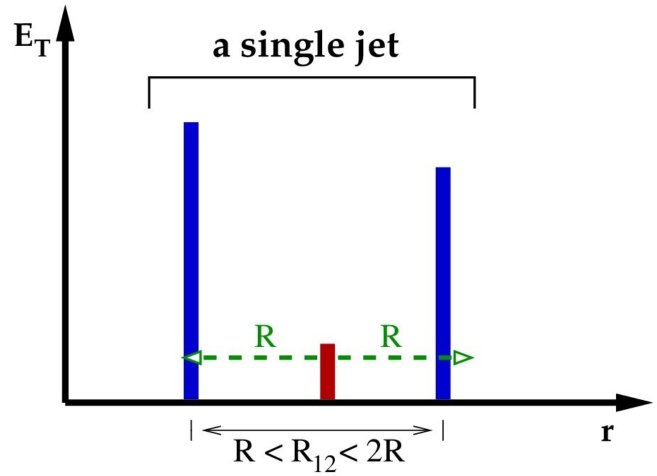
JETCLU is *not* infrared safe.

One possible dangerous configuration:



(Identified as 2 jets)

now add a soft gluon in the middle...



(Identified as 1 jet)

Jet Algorithms for Run II

The Jet Algorithm Working Group

As part of the *Run II Workshop on QCD and Weak Boson Physics* (1999–2000), the Jet Algorithms Working Group concentrated on the development of Run II jet algorithms that are:

- Free of theoretical and experimental difficulties
- Able to reproduce Run I measurements

Our principal goal was to develop a standard set of jet algorithms that are well-defined for both experiment (CDF & DØ) and theory.

The working group's recommendations are summarized in this presentation.

The full report is available as hep-ex/0005012.

Jet Algorithms for Run II

Theoretical Attributes

① *Infrared Safety* — The algorithm should find jets that are insensitive to soft radiation in the event.

② *Collinear Safety* — The algorithm should find jets that are insensitive to collinear radiation in the event.

③ *Boost Invariance* — The algorithm should find the same jets independent of boosts along the beam axis.

④ *Boundary Stability* — The kinematic variables that describe jets should be insensitive to the details of the final state.

⑤ *Order Independence* — The algorithm should be equivalent at the parton, particle, and detector levels.

⑥ *Straightforward Implementation* — The algorithm should be straightforward to implement in perturbative calculations.

Jet Algorithms for Run II

Experimental Attributes

① *Detector Independence* — There should be little or no dependence on detector segmentation, energy response, or resolution.

② *Minimization of Resolution Smearing* — The algorithm should not amplify the inevitable effects of resolution smearing and angle biases.

③ *Stability with Luminosity* — Jet-finding should not be strongly affected by multiple hard scatterings at high beam luminosities (*i.e.*, jets should not grow to excessively large sizes because of additional $p\overline{p}$ interactions).

④ *Resource Efficiency* — The jet algorithm should identify jets using a minimum of computer time.

⑤ *Reconstruction Efficiency* — The jet algorithm should identify all physically interesting jets (*i.e.* jets associated with partons).

(cont...)

Jet Algorithms for Run II

Experimental Attributes (cont.)

⑥ *Ease of Calibration* — The algorithm should not present obstacles to the reliable calibration of the final kinematic properties of the jet.

⑦ *Ease of Use* — The algorithm should be straightforward to implement with typical experimental detectors and data.

⑧ *Fully Specified* — All of the details of the algorithm must be fully specified, including specifications for clustering, definitions for energy and angles, and jet splitting/merging.

Jet Algorithms for Run II

Run II Recommendations

What were the working group's recommendations for Run II jet algorithms?

- *Midpoint Cone Algorithm*

  An improved version of the Run I cone algorithm, with fixes that satisfy theoretical issues. Midpoints between seed towers (and centers within groups of seeds) are included as additional starting locations for clusters.

- *Seedless Cone Algorithm*

  A cone algorithm that locates stable jet cones without introducing seed towers. The algorithm searches for stable cones starting with every calorimeter tower.

- K_T *Clustering Algorithm*

  A clustering algorithm that has been very successful at LEP and can be adapted for use at hadron colliders. The K_T algorithm behaves well theoretically and doesn't require ad hoc splitting/merging of cones.

Midpoint Cone Algorithm

Like the Run 1 cone algorithms, the midpoint algorithm is a seed-based algorithm. However, theoretical difficulties are fixed by the addition of *midpoints* in the list of seeds.

Steps:

① Generate p_T-ordered list of towers

② Find proto-jets around towers with $p_T >$ threshold

③ Generate midpoint list from proto-jets

④ Find proto-jets around midpoints

⑤ Go to the splitting/merging stage

To generate the midpoint list, it is sufficient to consider those midpoints where all seeds lie within a distance

$$\Delta R < 2.0 \cdot R$$

where R is the cone size.

Jet Algorithms for Run II

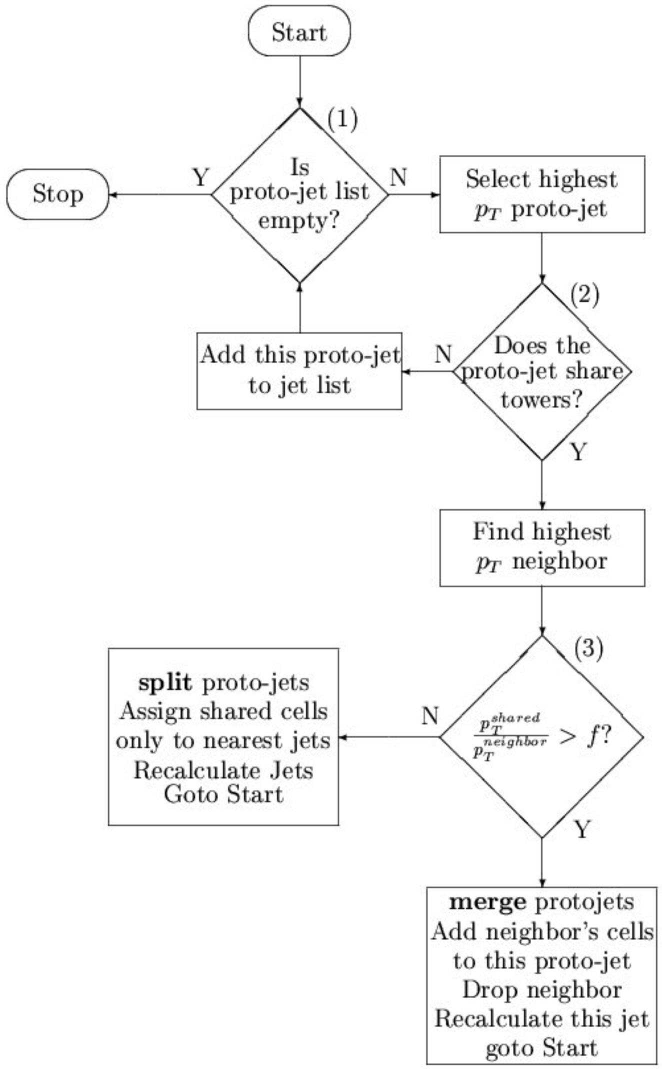
Midpoint Cone Algorithm

Splitting/Merging

- The splitting/merging stage of the jet algorithm dictates how to split or merge proto-jets with overlapping cones. This stage does not begin until all stable cones have been found.
- The splitting/merging algorithm always works with the highest p_T proto-jet on the list, and the list is reordered after each instance of merging or splitting.
- The decision to split or merge a pair of overlapping proto-jets is based on the percentage of p_T shared by the lower-p_T proto-jet.
- Proto-jets that share a fraction greater than f = 50% will be merged. Others will be split with the shared towers individually assigned to the proto-jet that is closest in η-ϕ space.
- At the end of splitting/merging, we are left with a list of jets.

Jet Algorithms for Run II

Splitting/Merging — Flowchart



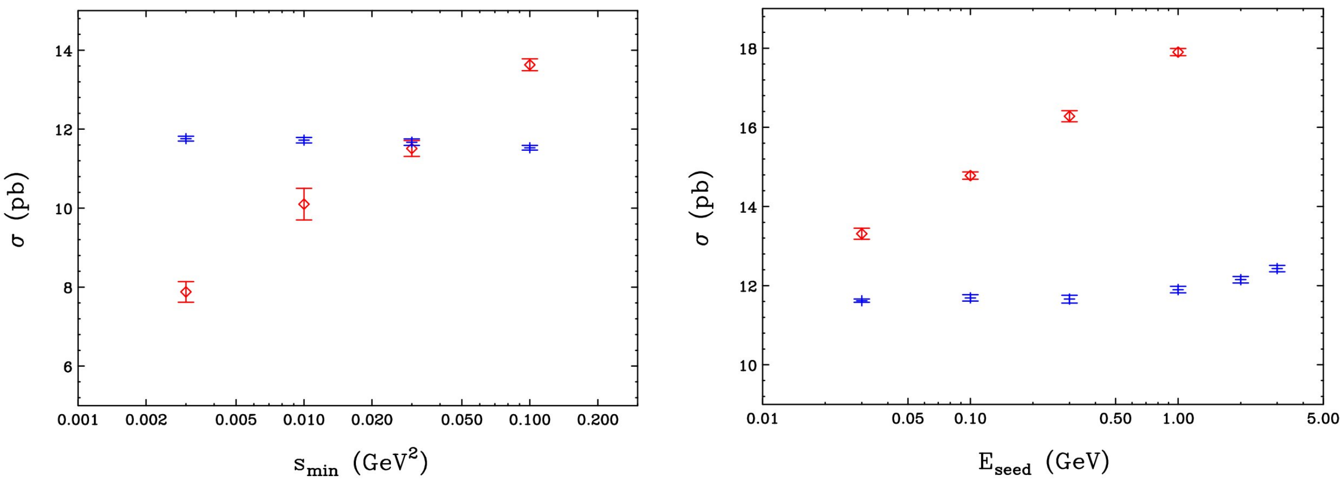
Use f = 50% !

Jet Algorithms for Run II

Midpoint Cone Algorithm

Monte Carlo Studies: DIS dijet cross section with midpoints and without midpoints.

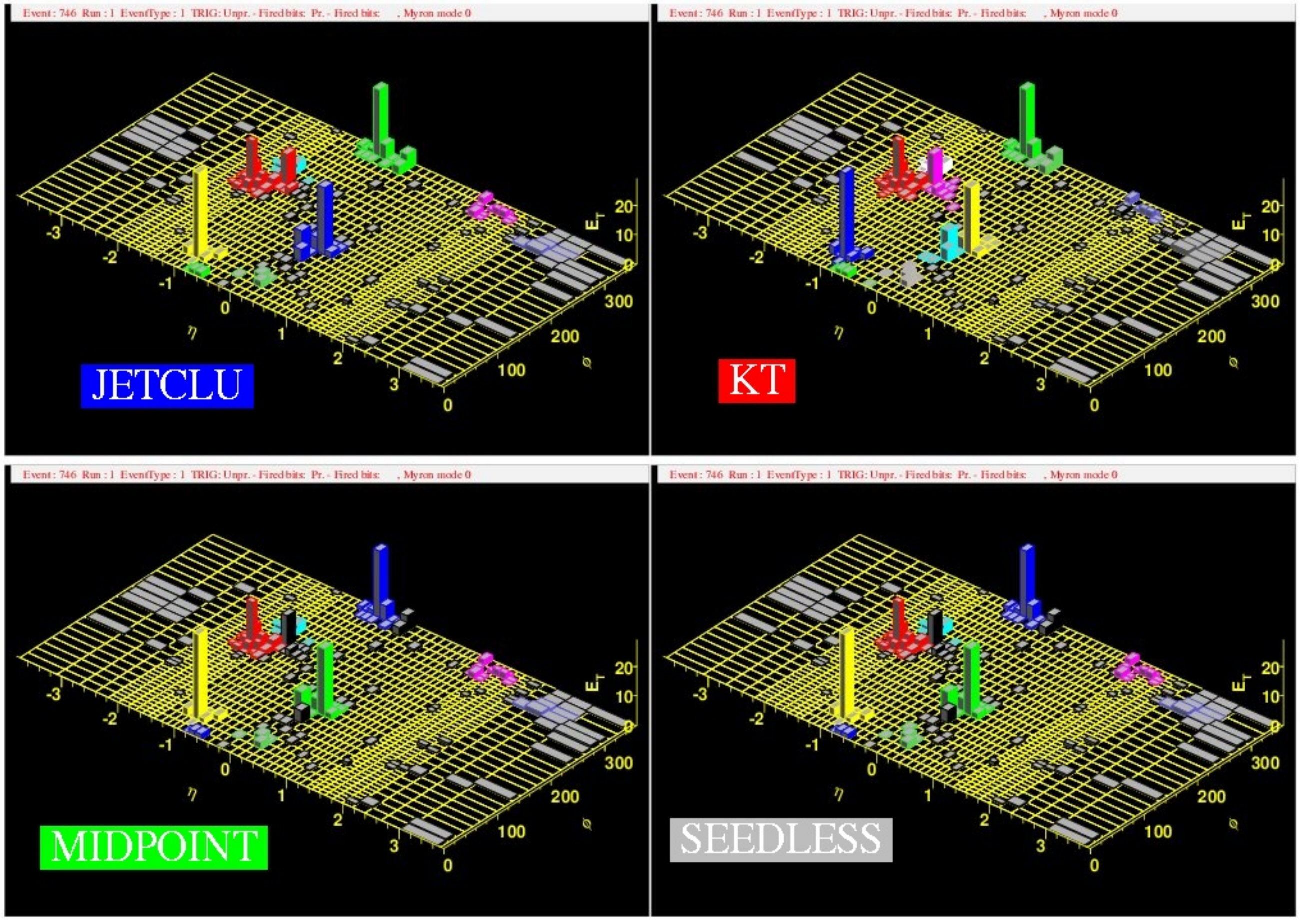
MEPJET ep Monte Carlo, $\sqrt{s}$ = 300 GeV, $Q^2 > 100$ GeV^2
Reconstructed jets are required to satisfy $E_T >$ 10 GeV, $-1 < y < 2$, $R_{jj} < 2$
Jets are reconstructed using the E-scheme, $R = 1.0$



The cross sections become stable *with* midpoints.

Jet Algorithms for Run II

The first interesting problem with the Midpoint Algorithm...



Midpoint & Seedless: Significant amounts of energy not collected into any jet!

Jet Algorithms for Run II

The first interesting problem with the Midpoint Algorithm... (cont.)

What is happening?

In some jet configurations, a high-E_T jet with one seed tower is close to a lower-E_T jet with another seed tower (yet $\Delta R > R_{\text{cone}}$).

An iteration process like the one in Midpoint or Seedless can lead to one stable cone centered only at the high-E_T jet, while the second jet — and thus a substantial amount of transverse energy — is ignored!

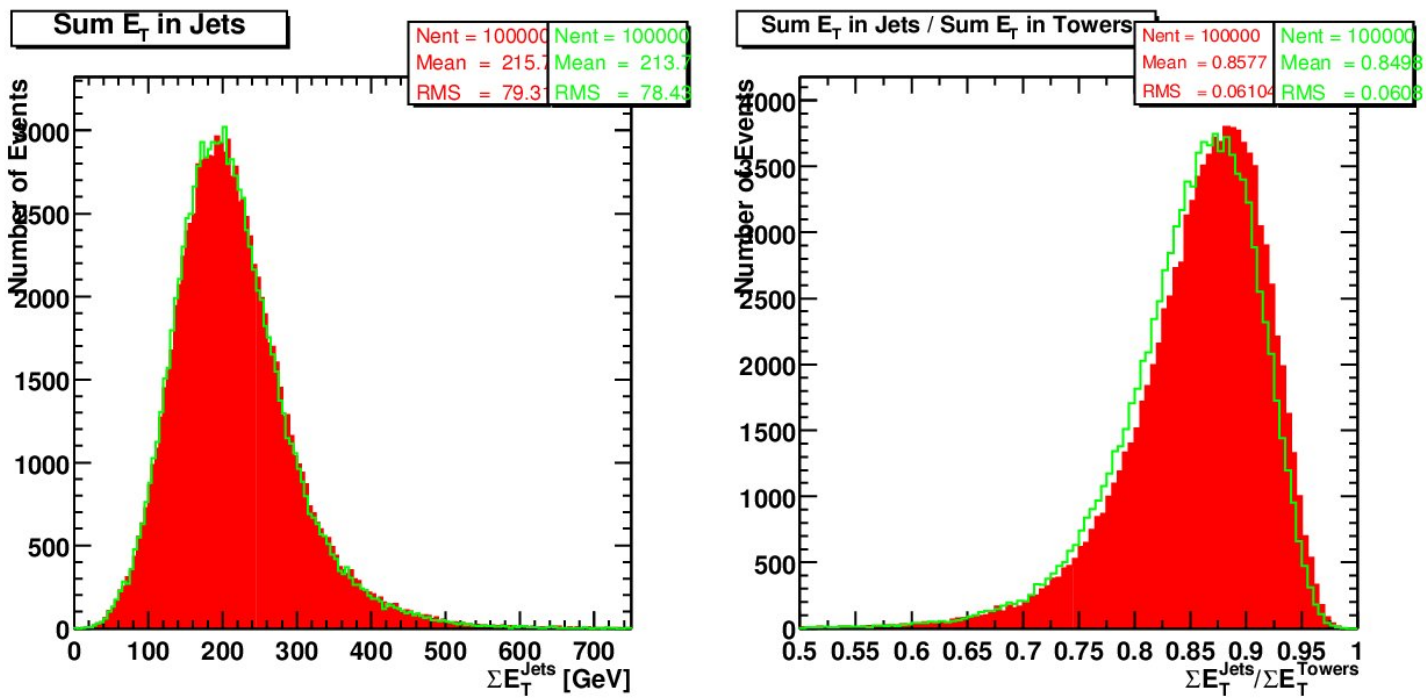
This phenomenon is not observed with JETCLU due to *ratcheting*, in which towers are never dropped during the iteration process.

In fact,

- The K_T algorithm is like JETCLU *with* ratcheting
- The Midpoint and Seedless algorithms are like JETCLU *without* ratcheting

Jet Algorithms for Run II

JETCLU vs. Midpoint for $t\bar{t}$ Monte Carlo Events (R = 0.4)

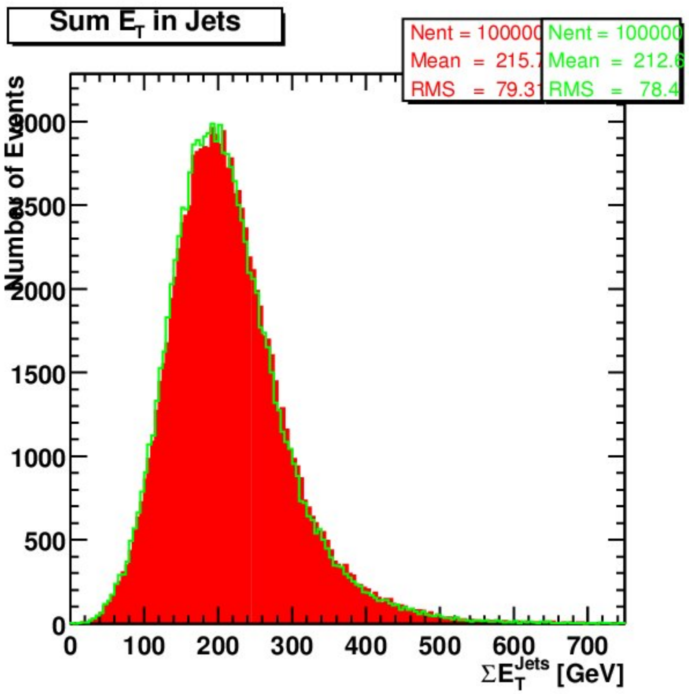


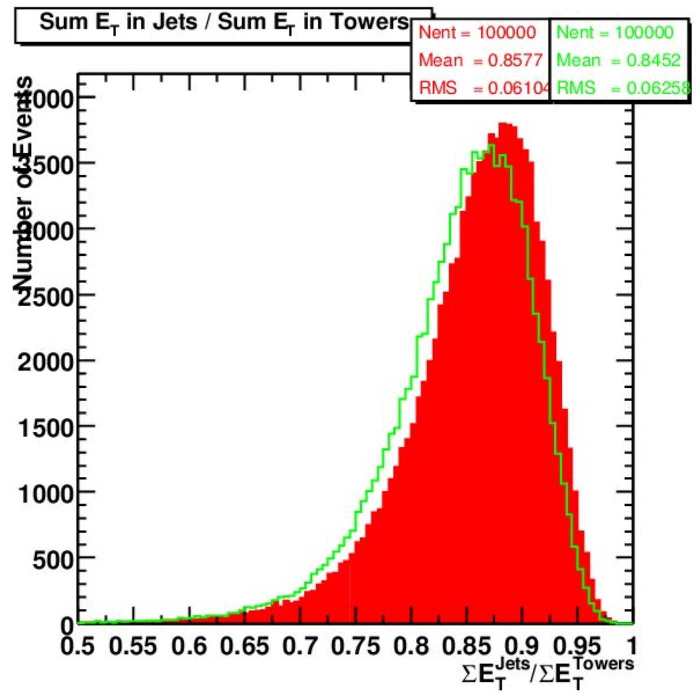
Red: JETCLU Green: Midpoint

Although the distribution of $\Sigma E_T(\text{jets})$ (left) is essentially identical, the distribution of $\Sigma E_T(\text{jets})/\Sigma E_T(\text{towers})$ (right) reveals that a smaller fraction of calorimeter E_T is clustered into jets for the Midpoint algorithm.

Jet Algorithms for Run II

JETCLU vs. JETCLU (No Ratcheting) for $t\bar{t}$ Monte Carlo Events (R = 0.4)

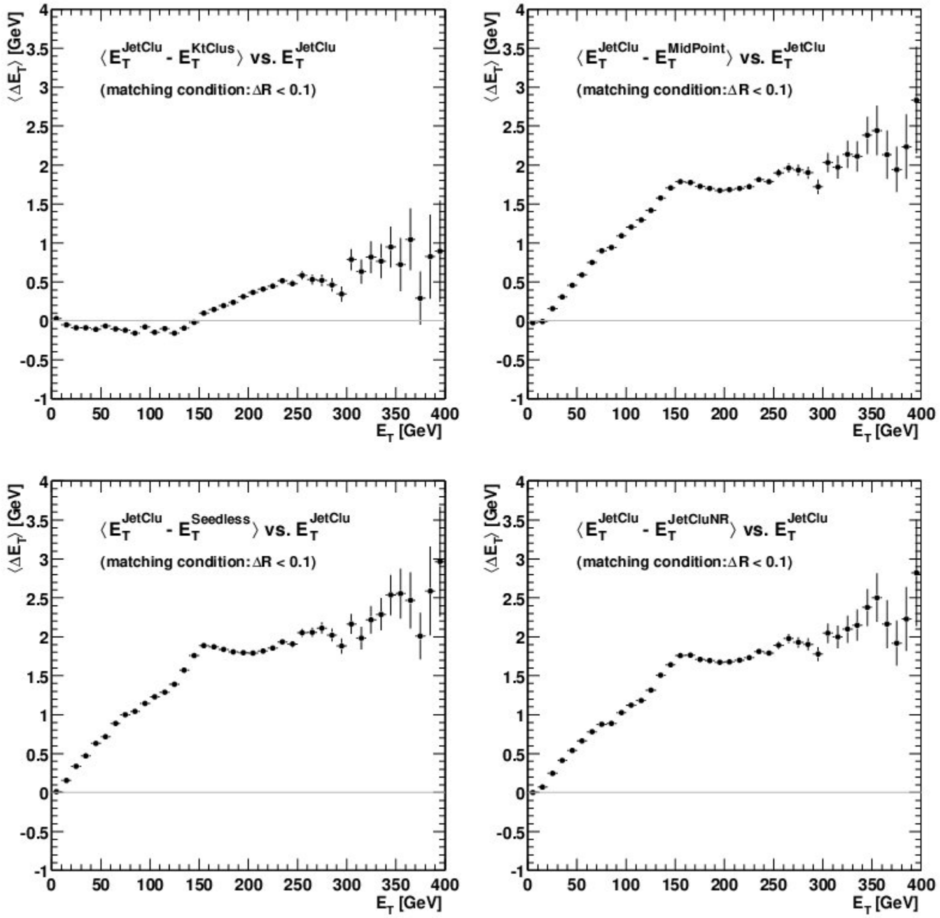




Red: JETCLU Green: JETCLU (No Ratcheting)

Jet Algorithms for Run II

Comparison of algorithms for HERWIG + CDF Detector Simulation (R = 0.7)
(Leading parton $E_T > 200$ GeV)



S. Ellis, J. Huston, M. Tönnesmann, hep-ph/0111434 (2001)

Jet Algorithms for Run II

Fix #1 to the Midpoint Algorithm

A major difference between the perturbative level (with a small number of partons) and the experimental level (with multiple hadrons) is the smearing that results from *perturbative showering* and *nonperturbative hadronization*.

Certain stable cone configurations, present at the perturbative level, can disappear from the analysis of real data due to the effects of showering and hadronization.

We have seen how cones *drift* during the iteration step of the Midpoint algorithms so that there are some seed towers that are not found in any jet.

The Fix:

> Use two values for the cone radius: one during search for stable cones ($R/2$) and the second during the calculation of the jet properties (R).

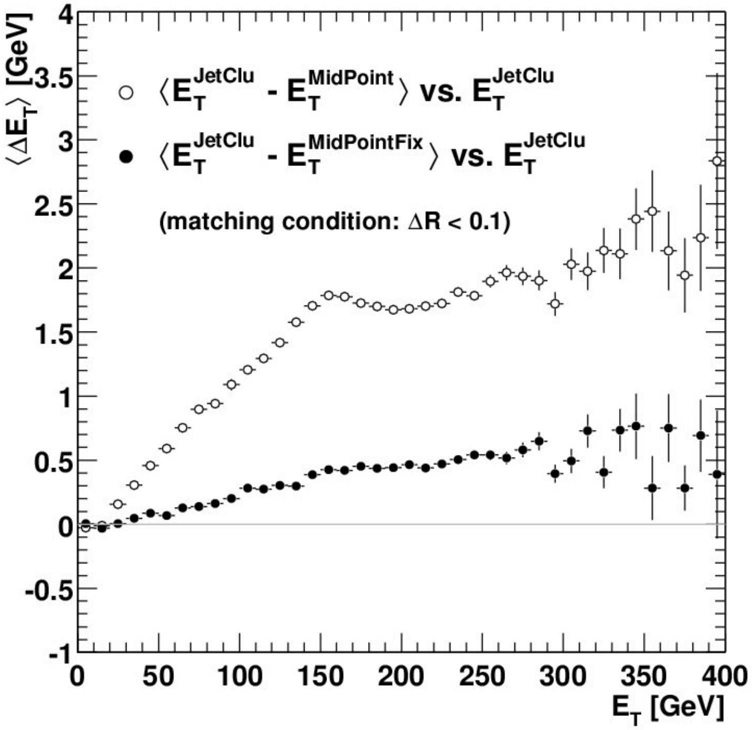
This initial search cone size $R/2$ was agreed upon on Dec 16, 2002 at a joint CDF/DØ/Theory meeting, and CDF has implemented it.

Jet Algorithms for Run II

Fix #1 to the Midpoint Algorithm (cont.)

Reducing the search cone size from R to $R/2$ reduces the influence of towers at the periphery (nearby high-E_T jets), placing more emphasis on the core of the cone.

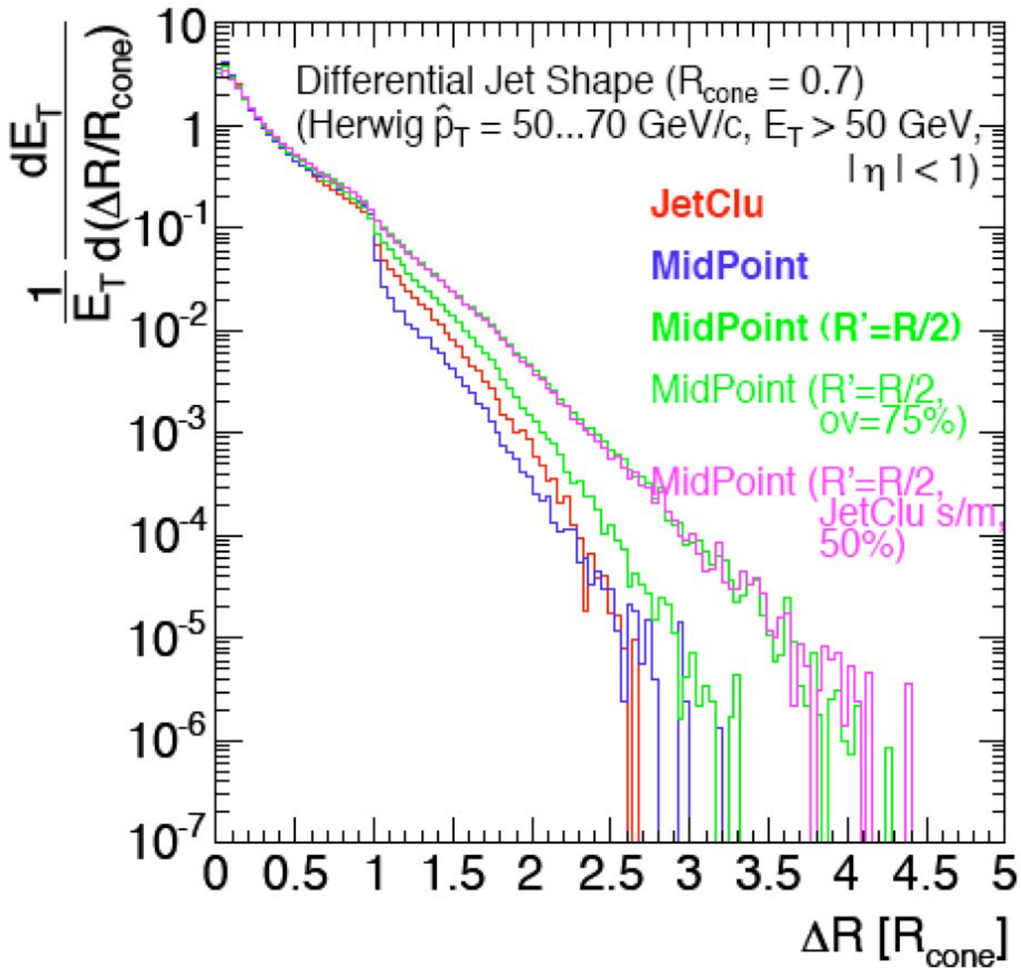
HERWIG + CDF Detector Simulation (R = 0.7):
(Leading parton $E_T > 200$ GeV)



The small-R fix removes the problem of the unstable low-E_T jet cone, but there are still occasions when smearing and splashout from the effects of showering and hadronization will lead to unstable middle cones.

Jet Algorithms for Run II

The second interesting problem with the Midpoint Algorithm...



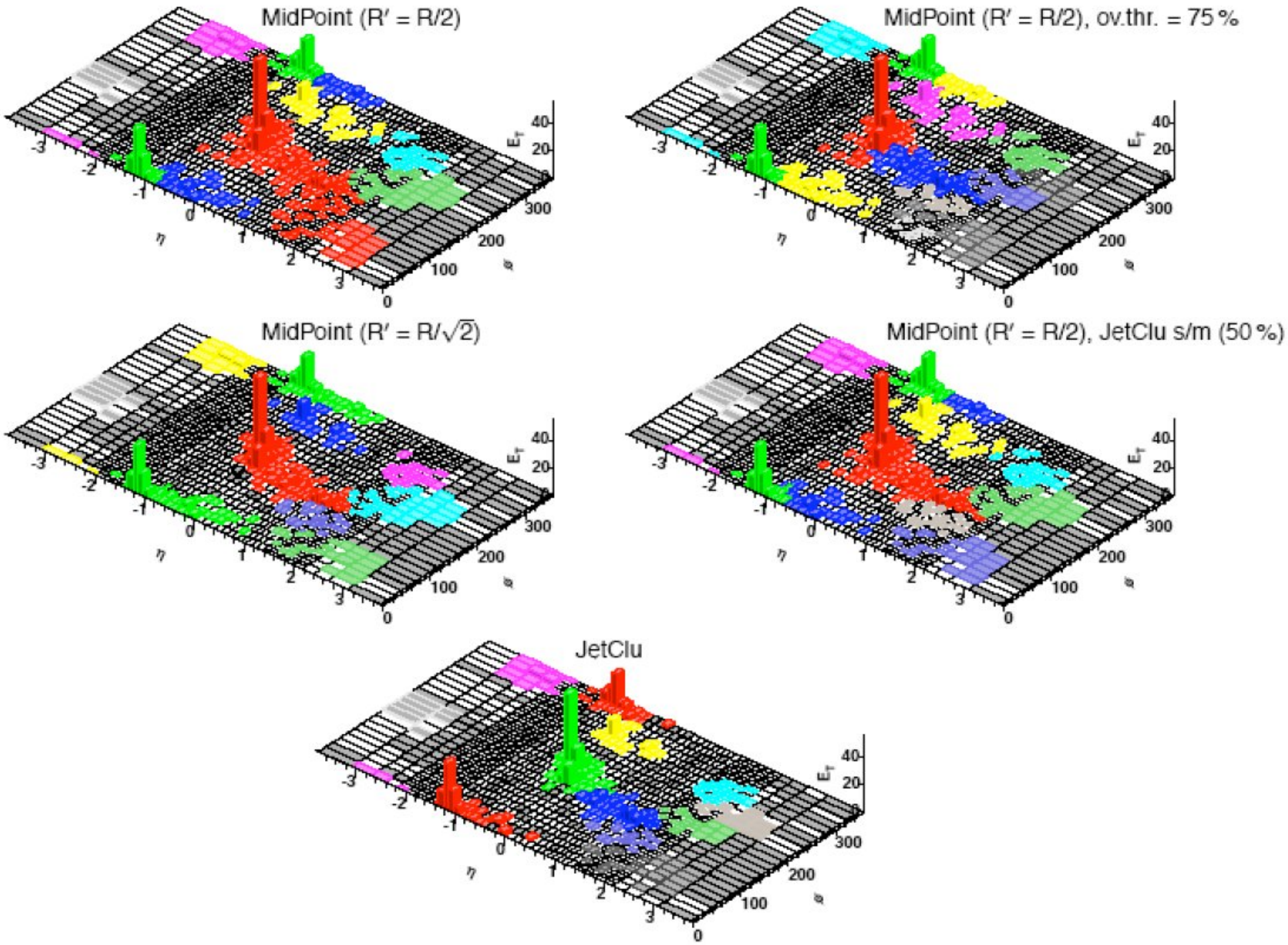
Midpoint jets are *fat!*

- Run 1: JETCLU, f = 75%, single merging iteration
- Run 2:
  - JETCLU, f = 75%, single merging iteration
  - Midpoint, f = 50%, multiple merging iterations

(Plot courtesy of M. Tönnesmann, J. Huston)

Jet Algorithms for Run II

CDF Run 151843 Event 1723497 (Jet100)



Jet Algorithms for Run II

Fix #2 to the Midpoint Algorithm

The choice to use $f = 0.5$ as the split/merge fraction for the Run 2 cone algorithms was rather arbitrary.

Studies have indicated that this the split/merge fraction is *the* sensitive parameter in the split/merge procedure. The size of jets are impacted significantly!

The Fix (22-OCT-2004):

A decision was made at CDF to switch the split/merge fraction to $f = 0.75$.
This is the same fraction used in JETCLU.

Using $f = 0.75$ is backward compatible with the Run 2 jet shape analysis.

It's not yet clear that $f = 0.75$ is the *optimal* value ... but it is an improvement over $f = 0.5$.

Jet Algorithms for Run II

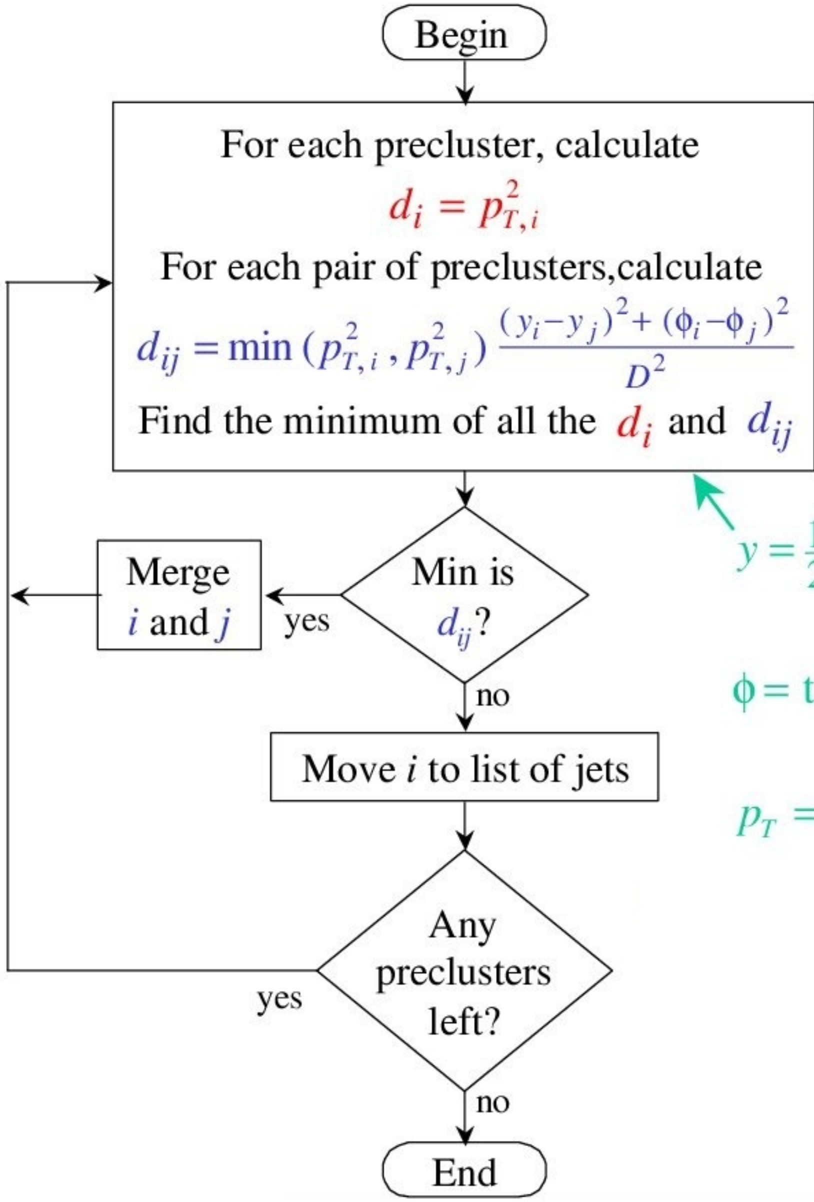
The K_T Jet Algorithm

The K_T *jet algorithm* successively merges pairs of partons, particles, or calorimeter towers in order of increasing relative transverse momentum.

The K_T algorithm typically contains a parameter D that controls the termination of the merging and characterizes the approximate size of the resulting jets.

Since the K_T algorithm fundamentally merges nearby particles, there is a close correspondence of jets reconstructed in a calorimeter to jets reconstructed from individual hadrons, leptons, and photons.

K_T Jet Algorithm for Run II



$$y = \frac{1}{2}\ln\frac{E + p_z}{E - p_z}$$

$$\phi = \tan^{-1}\frac{p_y}{p_x}$$

$$p_T = \sqrt{{p_x}^2 + {p_y}^2}$$

Merging scheme:

4-vector addition
(E-scheme)

$$p_x = p_{x,i} + p_{x,j}$$

$$p_y = p_{y,i} + p_{y,j}$$

$$p_z = p_{z,i} + p_{z,j}$$

$$E = E_i + E_j$$

The K_T Jet Algorithm

Features of the K_T algorithm:

- K_T jets are infrared and collinear safe (reliably calculable order by order in perturbation theory
- no overlapping jets (and no merging/splitting issue)
- every parton, particle, or tower is unambiguously assigned to a single jet
- no biases from seed towers
- true rapidity is used in the definition of d_{ij} to preserve boost invariance

The K_T algorithm is like the cone algorithm with a self-adjusting size (inspired by angular-ordered parton showering).

Jet Algorithms for Run II

Run II Jet Quantities (Midpoint, Seedless, K_T)

We use the following definitions for Run II jet quantities. For a jet 4-vector (E, p_x, p_y, p_z):

$$p = \sqrt{p_x^2 + p_y^2 + p_z^2} \qquad \theta = \cos^{-1}(p_z/p) \quad (0 \leq \theta \leq \pi)$$

$$p_T = \sqrt{p_x^2 + p_y^2} \qquad \phi = \tan^{-1}(p_y/p_x) \quad (0 \leq \phi < 2\pi)$$

which lead to

$$\begin{aligned} E_x &= E\cos\phi\sin\theta \\ E_y &= E\sin\phi\sin\theta \\ E_z &= E\cos\theta \\ E_T &= \sqrt{E_x^2 + E_y^2} = E\sin\theta \end{aligned}$$

By definition, $E = \sqrt{E_x^2 + E_y^2 + E_z^2}$.

In general, $E_T \neq p_T$, and they should not be used interchangeably.
The use of p_T is preferred!

Jet Algorithms at CDF

CDF has implemented the following algorithms for Run II:

① JETCLU (R = 0.4, 0.7, 1.0)

② Seedless Cone Algorithm (R = 0.4, 0.7, 1.0)

③ Midpoint Cone Algorithm (R = 0.4, 0.7, 1.0)

④ K_T Clustering Algorithm (D = 0.4, 0.7, 1.0)

For historical reasons, JETCLU is still widely used, although some collaborators are actively investigating Midpoint and K_T. We hope to move away from JETCLU in favor of the Run II algorithms.

The Seedless Algorithm has fallen out of favor because it is quite similar to Midpoint but much more CPU intensive. There have also been other problems...

* CDF has remained faithful to the Jet Algorithms Working Group specifications, except for the two fixes to Midpoint described in this presentation.

Summary

- In 1999–2000, the Jet Algorithms Working Group performed studies of jet algorithms and prepared a document (hep-ex/0005012) with jet algorithm recommendations for CDF, DØ, and future collider experiments.

  ① Seedless Cone Algorithm

  ② Midpoint Cone Algorithm

  ③ K_T Clustering Algorithm

  These new algorithms are intended to fix theoretical deficiencies in the Run I cone algorithms!

- Many studies have been performed – and continue to be performed – to test the Run II algorithms. The original Midpoint Algorithm suffered from a problem with cone instabilities, and CDF's solution solution was to use a smaller initial search cone size $R/2$ (hep-ph/0111434). This led to fat jets, which we are addressing by using split/merge fraction $f = 0.75$ rather than $f = 0.5$.

- CDF has adopted the Midpoint and K_T algorithms for Run II as prescribed by the Jet Algorithms Working Group. CDF also continues to implement its Run I jet algorithm JETCLU.

There's much more to do...and it will surely have a direct impact on the LHC!