Jet Algorithms

S. Catani and D. Zeppenfeld

1 Jet algorithms

Jet algorithms have the task to assign streams of hadrons in hard scattering processes to a jet, who's energy, mass and momentum can then be related to a collection of partons in a perturbative QCD calculation. Although, at the experimental level, jets can be defined by using rather general and intuitive procedures, if we would like to compute jet cross sections and properties by using QCD perturbation theory, the definition of jets should fulfil stronger constraints to guarantee its perturbative safety. Perturbative safety means that the definition has to be infrared safe (jet properties cannot depend on the presence of arbitrarily soft partons), collinear safe (jet properties cannot change by replacing a parton with a set of collinear partons carrying the same total momentum) and collinearly factorizable (jet properties should be insensitive to partons radiated collinearly to the beam direction). If the jet definition is not perturbative safe, we cannot perform calculations order-by-order in perturbation theory because they are affected by uncancelled infrared divergences. Of course, in the full QCD theory (i.e. beyond perturbation theory) the perturbative divergences are regularized by small physical cutoffs related to hadron masses and the finite experimental resolution (size of calorimeter cells, energy thresholds, etc.). The physical cutoffs are always present, independently of the jet definition. However, in the case of a perturbative safe definition, their effects are suppressed by some inverse power of the jet transverse energy E_T, and thus they can be made small by sufficiently increasing E_T. This power suppression is not at work in perturbative unsafe jet definitions, where the effects of the small physical cutoffs can amount to large corrections (of order unity) to the perturbative results. Thus, perturbative safe definitions are preferred.

Jet algorithms start from a list of "particles" which we would like to freely associate with calorimeter cells or hadrons at the experimental level, and with partons in a QCD calculation. Each particle i carries a 4-momentum p_i^μ, which we take to be massless. The task is to select a set of particles which are emitted close to each other in angle and combine their momenta to form the momentum of a jet. The selection process is called the "jet algorithm", the momentum addition rule is called the "recombination scheme".

Let us start with a discussion of recombination schemes. In a hadron collider environment the arbitrary boost of the hard scattering system along the beam axis needs to be taken into account in the definition of angles to ensure collinear factorizability. This is achieved by using

transverse momentum, $p_T = \sqrt{p_x^2 + p_y^2}$, rapidity $y = 1/2 \log(E + p_z)/(E - p_z)$ and azimuthal angle ϕ of the massless particles as the kinematic variables. When adding the massless 4-vectors of particles we obtain massive objects which only approximately correspond to the massless partons which we would like to associate with jets at tree level.

One popular choice, the Snowmass convention [1], leaves the question of jet mass open, by only defining total transverse energy, rapidity and azimuthal angle of a set of parton momenta, as the E_T weighted sums of the individual particle variables. For the original massless particles $E_{Ti} = p_{Ti}$ and $\eta_i = y_i$. The corresponding recombined variables for a cluster of particles are then given by the total transverse energy

$$E_T = \sum_i E_{Ti} \ , \tag{1.1}$$

the cluster pseudorapidity

$$\eta = \sum_i \frac{E_{Ti}}{E_T} \ \eta_i \ , \tag{1.2}$$

and the azimuthal angle of the cluster

$$\phi = \sum_i \frac{E_{Ti}}{E_T} \ \phi_i \ . \tag{1.3}$$

Note that the designation of η as pseudorapidity is purely conventional. It corresponds to neither the pseudorapidity nor the rapidity of the massive cluster and is approximately equal to either only in the limit of small cluster mass ($<< E_T$). The concomitant loss of Lorenz invariance is a serious disadvantage of the Snowmass convention. Another serious problem appears in resummation calculations (see Sect. 2): the kinematic boundary of jet E_T shifts (from $\sqrt{\hat{s}}/2$ in e.g. dijet kinematics) when including additional final state partons.

Because of these shortcomings we formulate all jet algorithms in the 4-momentum recombination scheme (also called E-scheme) in the following, i.e. the kinematic variables of a cluster of particles is given by direct addition of the 4-momenta of the individual massless particles:

$$p^\mu = (E, p_x, p_y, p_z) = \sum_i p_i^\mu \ . \tag{1.4}$$

Since the resulting clusters have a clearly defined mass, we must distinguish transverse energy E_T from transverse momentum p_T, and pseudorapidity η from rapidity y. We define

$$p_T = \sqrt{p_x^2 + p_y^2} \ , \qquad \phi = \tan^{-1} \frac{p_x}{p_y} \ , \qquad y = \frac{1}{2} \log \frac{E + p_z}{E - p_z} \ , \tag{1.5}$$

The rapidity y and azimuthal ϕ should be used as the legoplot position of the jet when calculating its separation from other particles or jets. Auxiliary quantities are

$$\theta = \cos^{-1} \frac{p_z}{\sqrt{p_x^2 + p_y^2 + p_z^2}} \ , \qquad E_T = E \ \sin\theta \ , \qquad \eta = -\log \tan \frac{\theta}{2} \ . \tag{1.6}$$

1.1 The k_T algorithm

The k_T algorithm [2] is a successive recombination algorithm. The idea is to recombine particles with nearly parallel momenta, beginning with the softest particles in the sample. This recombination stops once all clusters of particles are separated by a distance larger than D in the legoplot. The k_T algorithm starts from a list of protojets and their momenta, p_i^μ, which in the beginning consists of the list of all particles:

(1) For each protojet i define

$$d_i = E_{Ti}^2 \tag{1.7}$$

and for each pair of protojets i, j define a distance

$$d_{ij} = \min(E_{Ti}^2, E_{Tj}^2)\frac{(y_i - y_j)^2 + (\phi_i - \phi_j)^2}{D^2} \ . \tag{1.8}$$

(2) Find the smallest of all d_i and d_{ij} and call it d_{min}.

(3) If d_{min} is d_{ij} then merge protojets i and j to form a new protojet k of momentum

$$p_k^\mu = p_i^\mu + p_j^\mu \tag{1.9}$$

(4) If d_{min} is d_i then remove protojet i from the protojet list and move it to the list of completed jets.

(5) Continue with step 1 until the list of protojets is empty.

The algorithm is infrared safe because it renders all soft partons harmless: it either takes them off the protojet list or it combines them with nearby harder partons. The only effect of the soft parton then is a shift in the momentum of the recombined cluster. However, this shift is small, disappearing in the infrared limit, which guarantees infrared safety. Also for collinear emission the algorithm is safe, because in the limit of zero angle between two partons, these two will have the smallest d_{ij} and will thus be combined early on in the recombination process, thus restoring the momentum of the almost on-shell parton from which they originated by splitting. Finally, every original particle is assigned to exactly one jet, i.e. there are no splitting/merging issues to be resolved for the k_T algorithm.

1.2 ILCA: an infrared safe cone algorithm

Cone algorithms are intended to cluster all energy within a given radius, R, around a point in the legoplot, to form jets. Naively, this procedure is both infrared and collinear safe: the effect of infrared radiation on the cluster momentum vanishes in the infrared limit, and the energy measured for the jet is the same whether a single particle is at the core of the cone or whether there has been collinear splitting. This naive expectation can easily be violated, however, by the prescription for selecting cones. Two examples illustrate this point [3].

Assume that cones are constructed around actual energy depositions only. In Fig. 1(a) two particles are emitted at a distance greater than the cone radius R but smaller than $2R$ and therefore are assigned to separate cones, which are then identified as two distinct jets. The only difference in Fig. 1(b) is the emission of a third soft particle (a "soft gluon") between the original two particles. Now the additional cone around the soft energy deposition encompasses all three particles and they will be classified as a single jet. The presence of a soft particle changes the classification of a hard event: this is an example for an infrared unsafe algorithm. In perturbation theory, at sufficiently high order, an arbitrary number of soft gluons will be radiated, hence, cones should be allowed anywhere in phase space to anticipate this feature of higher order corrections. An arbitrary restriction on allowed cone positions may lead to an infrared unsafe algorithm.

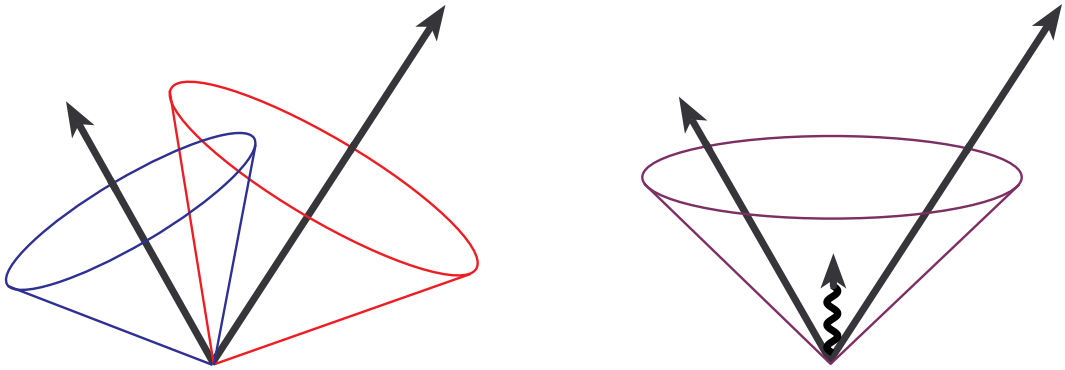

Figure 1: Example for a situation which can lead to infrared problems in an unsafe cone algorithm.

Similarly, an infinite number of collinear splittings occurs at higher order in perturbation theory. A possible collinear problem, resulting from E_T ordering of particles, is illustrated in Fig. 2. The difference between the two situations is that the central (hardest) parton may split into two almost collinear partons. On the left-hand-side the distance between the lateral partons is larger than R but the three hard partons all fall within a cone of radius R around the central parton, which happens to have the largest E_T. As a result, all three partons are recombined to a single jet. Collinear splitting renders the right hand parton to be the one with the largest E_T. Drawing the first cone around the highest E_T parton will recombine it with the two central partons and a separate jet is likely to be assigned to the remaining fourth parton. A differing jet number which depends on the presence or absence of collinear splitting must be avoided because the incomplete cancellation of the logarithmic divergences of real emission and virtual contributions will lead to a collinear unsafe jet algorithm. One can eliminate such ambiguities by making the selection or ordering of jet definition cones independent of the E_T of individual particles. Also, allowing trial cones anywhere in phase space would have made the two situations in Fig. 2 more similar: allowing cones centered between the central and the outside partons from the start would lead to a more similar jet

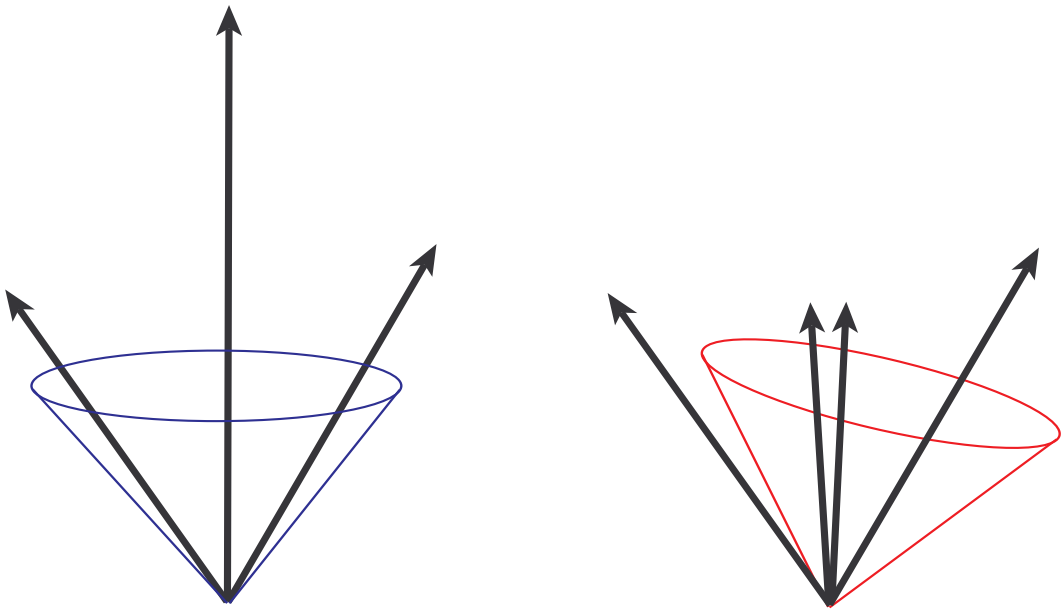

Figure 2: Example for a situation which can lead to collinear problems in an unsafe cone algorithm.

identification in the two cases.

The above considerations lead us to consider a cone algorithm which allows trial cones to be positioned anywhere in phase space, irrespective of the transverse momentum carried by individual particles or calorimeter cells. We start by formulating this seed-less algorithm at the calorimeter level, where the basic entities are calorimeter towers.

(0) Make a list of all calorimeter towers.

(1) Select the next tower on the list as the center of a trial cone of radius R.
Goto (4) if the list of towers is exhausted.

(2) Add the momenta of all towers inside the trial cone and determine the legoplot position (y, ϕ) corresponding to this momentum.

(3) If this position is outside the selected tower, discard the trial cone and go to (1).
If (y, ϕ) is inside the selected tower, add the set of towers inside the trial cone as a new entry to the list of protojets.

At this stage we have a list of protojets, and we need to split/merge them to make jets.

(4) Select the highest E_T protojet remaining on the list. (If the list is exhausted jet identification for the event is complete.)

(5) Does the selected protojet share any towers with other protojets?

(5.a) No: Move protojet to list of jets and continue with (4).

(5.b) Yes: Find the highest E_T protojet that shares towers with the selected protojet. (Call this the neighbor protojet.) Decide whether the E_T in the shared cells is greater than a fraction f of the E_T in the neighbor protojet.

(5.b.1) No: Split the shared towers.

(5.b.1.a) Allocate shared towers to either the selected or the neighbor protojet depending on which jet center is closer.

(5.b.1.b) Calculate the new momenta for the modified protojets, i.e. their E_T, and legoplot positions (y, ϕ). Continue with step (4).

(5.b.2) Yes: Merge the selected and neighbor protojets to form a new protojet. Add the momenta of both protojets and determine the total E_T and and the legoplot position (y, ϕ). Continue with step (4).

The procedure that defines the list of protojets is infrared and collinear safe. The additional steps completely define how to solve the problem of overlapping cones. The critical overlap fraction f is a free parameter of the algorithm and may be chosen as 50%, similar to the D0 choice in run I of the Tevatron (CDF uses 75%). The E_T-ordering of protojets in this split/merge step does not introduce collinear problems provided the cone size R is chosen sufficiently large.

The definition of calorimeter towers, i.e. a discretization of (y, ϕ) space, would be cumbersome in a theoretical calculation, and is indeed not necessary. In a perturbative calculation at fixed order, the maximal number, n, of partons is fixed. The only possible positions of stable cones are then given by the partitions of the n parton momenta, i.e. there are at most $2^n - 1$ possible locations of protojets. They are given by the legoplot positions of individual partons, all pairs of partons, all combinations of three partons etc. In a perturbative calculation, e.g. via a NLO Monte Carlo program, the protojet selection of the seedless algorithm (steps (0) to (3) above) can then be replaced as follows:

(0) Make a list of all possible cone centers. These are the legoplot coordinates of all parton momenta p_i, of all pairs of parton momenta $p_i + p_j$, of all triplets of parton momenta $p_i + p_j + p_k$, etc. For each cone center record which set of partons defines it.

(1) Select the next cone center on the list as the center of a trial cone of radius R. Goto (4) if the list of cone centers is exhausted.

(2) Add the momenta of all partons inside the trial cone and determine the legoplot position (y, ϕ) corresponding to this momentum.

(3) If this position is different from the trial cone center, i.e. if the cone center record and the list of partons inside the trial cone disagree, discard the trial cone and go to (1). If (y, ϕ) is the trial cone center, add the set of partons inside the trial cone as a new entry to the list of protojets.

As before, different protojets may share partons, i.e. they may overlap. The required split/merge step is then identical to the calorimeter level steps (4) and (5), with towers replaced by partons as elements of protojets.

In an actual experiment the number of calorimeter towers may be very large (order 6000 for tower sizes of $\Delta\eta \times \Delta\phi = 0.1 \times 0.1$ and an η coverage of ± 5 units of pseudorapidity). The calorimeter level algorithm may then be rather slow computationally. The question arises whether an acceptable approximation of the seedless algorithm can be constructed, analogous to the parton level short-cut, by considering only those towers which have energy depositions above a minimal seed threshold. One would like to replace the list of parton momenta above by the list of tower momenta with

$$p_{Ti} > E_{T,seed} \, . \tag{1.10}$$

Since the algorithm is infrared and collinear safe when $E_{T,seed} = 0$, it is always possible to chose the seed threshold $E_{T,seed}$ low enough so that variations of $E_{T,seed}$ lead to negligeable variations in any observable under consideration.

One would like to include in the determination of jet momenta all towers, of course, which lie inside the cone of radius R around the protojet axis. This requires an additional iteration of the cone axis in the parton level algorithm when a seed threshold is imposed. The steps leading to the definition of protojets can then be modified as follows:

(0) Make a list of all possible cone centers. These are the legoplot coordinates of all parton/tower momenta p_i with $p_{Ti} > E_{T,seed}$, of all pairs of such parton/tower momenta $p_i + p_j$, of all triplets $p_i + p_j + p_k$, etc.

(1) Select the next cone center on the list as the center of a trial cone of radius R. Goto (3) if the list of cone centers is exhausted.

(2) Add the momenta of all partons/towers inside the trial cone (also those with $p_{Ti} < E_{T,seed}$) and determine the legoplot position (y, ϕ) corresponding to this cone momentum. Use (y, ϕ) as the new center of the trial cone and iterate this step until the position is stable. The set of all towers/partons inside the final trial cone constitutes a new protojet. Continue with step (1).

(3) Eliminate all duplicate protojets, i.e. protojets with an identical set of towers/partons.

With these changes, the resulting algorithm (named Improved Legacy Cone Algorithm or ILCA) is quite close to those used in run I of the Tevatron. The main change is the inclusion of midpoints of seeds (the $p_i + p_j$ pairs) and of centers of larger numbers of seeds as additional seed locations for trial cones. Including these additional midpoints is absolutely crucial in perturbative calculations in order to achieve infrared safety (see discussion on Fig. 1). When dealing with data, these effects are somewhat diminished, because with sufficiently low seed thresholds $E_{T,seed}$, a large number of trial cones will be generated from actual soft energy depositions in the calorimeter. However, because these soft energy depositions will decide how many jets are reconstructed, one potentially introduces a high sensitivity of jet

observables to soft hadrons, Monte Carlo modelling of soft particles etc. The inclusion of the extra midpoints eliminates these soft effects because observables no longer depend on whether soft emission actually took place.

2 Resummed calculations

The ILCA and k_T-algorithm eventually lead to jets whose topology is not extremely different from that expected on the basis of a naive definition in terms of cones in azimuth-rapidity space. This is obviously true for the ILCA, where jets can contain particles whose distance is smaller than $2R$ and have a shape that differs from a cone-shape only because of the merging/splitting procedure. In the k_T-algorithm, jets have no sharp boundaries, but opening angles of particles within each jet are, typically, smaller than D and all opening angles between jets are larger than D. The detailed jet structure is, however, different in the two algorithms. Although both algorithms are perturbative safe, the differences show up in higher-order perturbative calculations.

Higher-order perturbative computations and, in particular, resummed calculations can be necessary in special kinematics configurations that lead to large logarithmically-enhanced contributions at any fixed order in perturbation theory. Typical examples are calculations of jet cross sections near the phase-space boundary and of the fine internal structure of jets (shape variables, subjets, etc.). These quantities can be strongly dependent on the jet definition. The corresponding perturbative calculations do strongly depend on the jet definition, because they are the result of the integration of the QCD matrix elements (which do not depend on jets) over phase-space regions whose boundaries depend on the fine details of jet kinematics.

As an example of this strong sensitivity, we can consider the one-jet inclusive cross section as a function of the transverse momentum p_T of the jet. *If* the jet variable p_T is defined by using the 4-momentum recombination scheme (see Eqs. (1.4)–(1.6)), the kinematical boundary is $x_T \leq 1$, where $x_T = 2p_T/\sqrt{S}$. Close to the boundary $x_T \sim 1$, the perturbative contributions are enhanced by large logarithmic corrections $(\alpha_S \log^2(1-x_T))^n$ that need to be resummed to all orders in α_S. Techniques to perform this resummation can be developed (see below). However, *if* the jet variable p_T is defined by using a recombination scheme that does not conserve the 4-momentum (e.g. the true p_T in Eq. (1.5) is replaced by the variable E_T in Eq. (1.1)), the kinematical boundary for x_T, although close to $x_T = 1$, is not fixed: the corresponding large logarithms cannot be resummed because the x_T-boundary shifts in a complicated manner depending on the number of final-state partons in the calculation [4].

The feasibility of resummed calculations depends not only on the recombination scheme but also on the jet algorithm, as is well known for jets in e^+e^- annihilation [5]. The jet definition of the k_T-algorithm is inspired by the parton shower picture of jet fragmentation [2]. Thus resummed calculations can be carried out by using the analytic version of the recurrence techniques used to generate multiparton final states in Monte Carlo parton showers. This is demonstrated by explicit calculations of subjet multiplicity and rates in hadron collisions [6]. The ILCA has still to be investigated in this respect. The two algorithms differ only slightly

at highly inclusive level [2]. Thus, in these cases (such as the large-x_T behaviour of the one-jet inclusive cross section), resummed calculations in the ILCA should be feasible as in the k_T-algorithm. Studies of the internal structure of ILCA jets may instead be more difficult since it depends on the procedure to merge/split overlapping jets.

3 Conclusions

During the Les Houches workshop discussions were centered on the general properties of jet algorithms, in particular their infrared and collinear safety at the perturbative level. The k_T-algorithm and the seedless cone algorithm described in Section 1.2 fulfil these requirements. Beyond these theoretical concerns there are many experimental issues which need to be addressed to obtain a practical algorithm. Among these are ease of energy calibration, effects of underlying event and overlapping events in a high luminosity hadron collider environment, high jet reconstruction efficiency, and efficient use of computer resources in reconstructing jets. These issues have been addressed in a parallel study, during the Run II QCD Workshop at Fermilab [7]. In particular it has been shown that the ILCA produces small corrections only, when compared with the jet algorithms used in run I of the Tevatron. We refer the reader to the Proceedings of the Run II Workshop for a detailed study of these effects.

References

[1] J. Huth et al., Proc. of the 1990 DPF Summer Study on High Energy Physics, Snowmass, CO, ed. by E. L. Berger (World Scientific, Singapore, 1992), p.134.

[2] S. Catani, Y. L. Dokshitzer, M. H. Seymour and B. R. Webber, Nucl. Phys. **B406**, 187 (1993); S. D. Ellis and D. E. Soper, Phys. Rev. **D48**, 3160 (1993).

[3] M.H. Seymour, Nucl. Phys. **B513**, 269 (1998).

[4] S. Catani and B. R. Webber, JHEP **9710**, 005 (1997) [hep-ph/9710333].

[5] N. Brown and W. J. Stirling, Phys. Lett. **B252**, 657 (1990); S. Catani, Y. L. Dokshitzer, M. Olsson, G. Turnock and B. R. Webber, Phys. Lett. **B269**, 432 (1991).

[6] M. H. Seymour, Phys. Lett. **B378**, 279 (1996) [hep-ph/9603281]; J. R. Forshaw and M. H. Seymour, JHEP **9909**, 009 (1999) [hep-ph/9908307].

[7] G. C. Blazey et al., *Run II Jet Physics*, contribution of the "Jet Physics" group to the proceedings of the workshop on *QCD and Weak Boson Physics* in preparation of Run II at the Fermilab Tevatron, March - November, 1999.