Journal of Global Optimization **9**: 41–64, 1996.

Tracking Elementary Particles near their Primary Vertex: A Combinatorial Approach

JEAN-FRANÇOIS PUSZTASZERI jfp@aloha.cern.ch
ECP Division, CERN, CH-1211 Geneva 23, Switzerland

PAUL E. RENSING rensing@aloha.cern.ch
PPE Division, CERN, CH-1211 Geneva 23, Switzerland

THOMAS M. LIEBLING liebling@dma.epfl.ch
Département de Mathématiques, EPFL, CH-1015 Lausanne, Switzerland

Received March 1, 1996

Abstract. Colliding beams experiments in High Energy Physics rely on solid state detectors to track the flight paths of charged elementary particles near their primary point of interaction. Reconstructing tracks in this region requires, per collision, a partitioning of up to 10^3 highly correlated observations into an unknown number of tracks. We report on the successful implementation of a combinatorial track finding algorithm to solve this pattern recognition problem in the context of the ALEPH experiment at CERN. Central to the implementation is a 5-dimensional axial assignment model (AP5) encompassing noise and inefficiencies of the detector, whose weights of assignments are obtained by means of an extended Kalman filter. A preprocessing step, involving the clustering and geometric partitioning of the observations, ensures reasonable bounds on the size of the problems, which are solved using a branch & bound algorithm with LP relaxation. Convergence is reached within one second of CPU time on a RISC workstation in average.

Keywords: Integer Programming, Multiple-Target Tracking, Extended Kalman Filtering, Particle Physics, Pattern Recognition

1. Introduction

A large part of the research activities currently taking place in High Energy Physics is devoted to the study of fundamental interactions produced by colliding beams of elementary particles. Electron-positron colliders, such as the LEP storage ring located at the European Laboratory for High Energy Physics (CERN) in Geneva, operate at the resonance of the Z^0 gauge boson, which decays into an a priori unknown product. The product interactions, or events, are viewed and analyzed via the means of large composite detectors generally made of concentric cylindrical shells of electronic arrays occupying a volume of a few thousand cubic meters. The outer shells are made of calorimeters, which are designed to absorb incoming particles and measure their deposited energy, while the inner part is composed of ionization chambers and solid-state devices whose purpose is to record point-like observations on the flight path, or track, of every charged particle produced by the

Z^0 decay. Tracking is defined as the reconstruction of the flight path of particles from space-point data produced in this region.

A particle is identified only once its track has been reconstructed. The reconstructed tracks are used to determine the topology of the event, and in particular to calculate the exact location of the point of origin, or vertex, of each particle. Most product particles have their origin at the primary point of interaction (the primary vertex), but others are themselves the product of mother particles, and their vertex is located elsewhere in the detection volume. Measuring the vertices of particles with precision is of prime importance to the experimental analysis of fundamental physics interactions. As the *precision* of tracking has a direct bearing on this analysis, it represents a crucial step in the processing of High Energy Physics data.

2. Conventional Tracking Algorithms

The problem defined above belongs to the class of Multiple-Target Tracking problems (MTT) which are have been studied extensively in the field of information processing. MTT instances arising from surveillance applications and computer imaging are routinely being solved in near real-time conditions with the help of general purpose algorithms, such as the Multiple Hypothesis Tracking (MHT) method, originally developed by Reid [26], and the Joint Probabilistic Data Association Filters (JPDAF) by Bar-Shalom *et al.*, [2]. These methods rely on the ordering of observations in a scan sequence, and proceed by constructing the full tree of track hypotheses, with each branch representing the association of data in adjacent, or near-adjacent, scans. These methods are suboptimal in the sense that they rely on heuristics to limit the number of branches in the tree, or to normalize the hypotheses at every scan. Several variants of these methods have been proposed in the literature (for a review of mainstream MTT algorithms, see Blackmann [4] for instance).

The High Energy Tracking problem differs from the mainstream MTT applications in that, due to the very short lifetime of the product particles (a few nanoseconds) and to the a priori unknown event topology, it may be difficult to define a reasonable scan sequence if the local density of observations is large. This prohibits, at least locally, the use of conventional sequential methods. By the same token, data are available all at once, which renders a global treatment of information (or batch processing) an interesting alternative to the scan-by-scan approach, an idea which is developed in this paper. Combinatorial algorithms to solve the generalized MTT data association problem have been proposed by Poore [22], and Poore and Rijavec [23], [24]. In the latter, the MTT problem is modeled as a multi-dimensional assignment problem which is solved by Lagrangian relaxation. For a complete coverage of multi-dimensional assignment problems, see Pardalos, Pitsoulis and Resende [17], and Pardalos and Wolkowicz [18].

Attempts have already been made to solve the High-Energy Physics MTT problem using batch processing methods (albeit adaptive ones). Peterson [21], Stimpfl

and Garrido [28], among others, proposed a Hopfield-type neural network approach, while Gyulassi and Harlander [12] used a generalization of the Radon transform to derive an elastic tracking algorithm relying on a lexicographic search for deformable "template" tracks. A hybrid algorithm based on simulated annealing and making use of a mean-field approximation on the state of indicator variables was developed by Peterson and Anderson [20]. Practical implementation of these methods have often revealed performances equal to simpler nearest-neighbour search, "road-finding" and tree algorithms [8], which has hampered somewhat their more widespread use in running experiments to this day. This situation may change with the next generation of particle colliders and detectors, and the far more complex instances of tracking problems associated with them.

While the nearest-neighbour algorithms are efficient in regions where tracks are well separated, they are often used across the entire detection region regardless of the local density of observations which altogether tends to limit their effectiveness. The following section describes the tracking environment of the ALEPH High Energy Physics experiment at CERN, and the conventional algorithms used for data reconstruction there.

3. Problem Description

ALEPH is one of the four detectors located on the Large Electron-Positron Collider (LEP) at CERN. Its principles of operations are fully described in Decamp *et al.* [11], and a schematic view of its cylindrical tracking components is given in Figure 1a. The assembly is centered along the beam axis and about the beam spot, the region where the two beams collide. The tracking environment differs substantially from one component to the next: the outer shell is a gas-filled Time Projection Chamber (TPC) operating on the principle of measuring the drift distance of ionization. The chamber produces three-dimensional points which are used for tracking. The Inner Tracking Chamber (ITC) is a proportional wire drift chamber used mainly for triggering purpose, which doubles as a tracking device. The uncertainty of observations in the direction of the beam axis are too large to be used in practice, so two-dimensional observations, distributed over nine layers, are available for tracking in this region.

The innermost device is a solid-state silicon strip vertex detector (VDET), made of two layers of overlapping silicon wafers with an inlay of orthogonal aluminum strips (Figure 1b). As a charged particle traverses a wafer, the ionization charge it produces in the silicon is picked up by the nearest pair of orthogonal strips (Figure 1d). The signals induced in the strips are used to identify the orthogonal pair, from which a three-dimensional point (knowing the position of the strips on the wafer) can be reconstructed. This device can be pictured as four copies of a one-dimensional detector embedded in a three-dimensional framework described by an orthogonal local coordinate system, the "z" direction used for observations which lie perpendicular to the beam axis, and $\rho - \phi$ for observations which lie parallel to it.

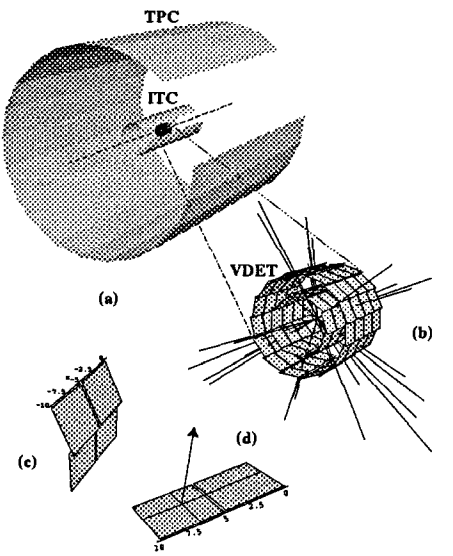

Figure 1. ALEPH tracking chambers (a). The VDET is made of two concentric layers of silicon wafers (b) centered at the origin. Wafer overlap is shown in (c). A VDET module is a pair of wafers (d) which triggers two orthogonal signals when a charged particle traverses it (scale in cm)

All three devices are immersed in a magnetic field of constant magnitude along the beam axis, which constrains charged particles to follow a helical path about that axis. The energy of the particle is proportional to the radius of curvature of its track.

3.1. Algorithm Implementation Issues

The tracking detectors operate on very different scales and resolutions. A charged particle easily travels more than a meter in the TPC and generates up to twenty-one space-points with a resolution of $180\mu m$. By contrast, the same particle generates only two space-points in the vertex detector, but with a resolution which is better by an order of magnitude. Unless a particle traverses an overlap region of the vertex detector, the latter does not produce sufficient information by itself to reconstruct a helix. Track reconstruction in the important vertex detector region relies therefore entirely on outer tracking.

There are also major differences in the density of observations between different regions: most tracks have their origin within the volume bounded by the vertex detector, and are therefore well separated by the time they enter the TPC region. Conversely, the density of information is the greatest in the vertex detector, where local track quality matters the most.

A third difficulty inherent to High Energy Physics tracking is the presence of non-negligible local deviations from the ideal helical flight path of particles due to

interactions of the latter with matter present in the detector. Most noticeable are *multiple scattering* effects, described in detail in Scott [27]. Only approximate models have been proposed to calculate the deflection angle due to multiple scattering, which may be viewed as a local change of state in the helical model. This effect is lessened for highly energetic tracks. The implementation of an seamless tracking algorithm (global or sequential) applicable to all regions at once is therefore difficult.

3.2. Track Reconstruction

Track reconstruction is first performed in the outer region by means of sequential road-finding methods, the implementation of which is described in Comas *et al.* [10]. In order to limit multiple scattering effects, it proceeds by identifying tracks segments with the smallest curvature, and therefore the highest energy, which are fitted by least-square methods and extrapolated inwards in the direction of the vertex detector. Given the good track separation in these regions, this method performs generally quite well locally [7].

When it comes to actually associate outer tracks with the more accurate vertex detector observations, the step above should only be viewed as a partitioning of outer observations into partial tracks. In order to calculate track parameters accurately, multiple scattering effects at the detector boundaries and in the silicon wafers of the vertex detector need to be taken into account. A more accurate fitting procedure, based on an extended Kalman filter, is activated for this purpose. Its implementation is described in the next section, and the prime advantage of the procedure is to account for low-angle multiple scattering effects, a model of which is directly introduced in the covariance matrix of the process noise.

Figure 2 shows two outer tracks, their 5-standard deviation extrapolation area represented by an elliptical conic road around the track, covering a number of orthogonal vertex detector observations to which the tracks can be assigned. If the optimality criteria for application of a Kalman filter (i.e., a linear transport equation, normal distribution of measurement and process noise, mutual independence of time series) were truly satisfied, then an optimal association of observations could be made sequentially based on some arbitrary track ordering, a principle relied upon entirely in the ALEPH tracking software up to now.

3.3. Kalman Filter Implementation

Extended Kalman filtering methods have been described extensively in the literature (see Catlin [9] for instance), and only a brief description of implementation details in the specific context of this work is given here (for a full description, see Comas *et al.* [10]). A damped regular helix coiling around the beam axis models the particle path in the detector medium. The state equation is represented by the

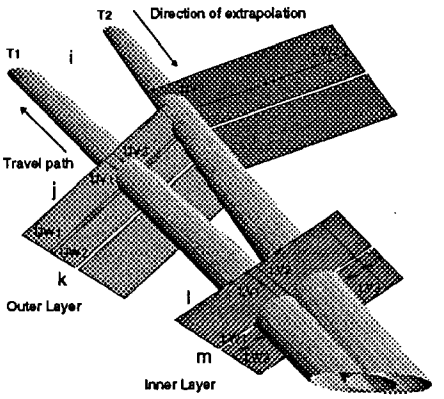

Figure 2. Two partial tracks extrapolated from the outer region into the vertex detector, together with their five standard-deviation extrapolation errors. A possible assignment of observations is indicated by matching indices.All observations are correlated in terms of track assignment

five-vector

$$\vec{x} = (u(r), z(r), \Phi(r), \lambda(r), \omega(r))^T$$

with parameters representing the xy and z coordinates of the observations on the vertex detector wafer, the angle of the xy projection of the track with respect to the x-axis, the angle of incidence, and the curvature of the track respectively. The radius r is the Euclidean distance between the origin and a point on the projection of the helix in the xy-plane. The transport equation $\vec{f}_k(\vec{x}_k)$ of the state vector $\vec{x}_k$ is given in the Appendix. It is nonlinear (hence the extended approach), and the filtered estimator (or more precisely the conditional mean) of the Kalman filter can no longer be calculated analytically. Because of computational constraints, only a first-order Taylor series expansion of the transformation is used. The system equation

$$\vec{x}_{k+1} = \vec{f}_k(\vec{x}_k) + \vec{w}_k$$

contains the process noise $\vec{w}_k$ modeling multiple scattering effects described earlier. The measurement equation is given by

$$\vec{m}_k = H\vec{x}_k + \vec{\epsilon}_k$$

where $\vec{\epsilon}_k$ is the measurement error vector of the actual observations $\vec{m}_k = (u_k, z_k)^T$. The transport function for the measurement, H, is in our case a matrix of constants

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

The covariance matrix Q of the process noise contains a description of multiple scattering, and is obtained by solving the differential equation

$$\frac{dQ}{dl} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \cdot Q(l) + Q(l) \cdot \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \cdot \frac{d\Theta^2}{dl} \tag{1}$$

over a set of boundary conditions determined by the type of material present in the path segment of length l. Θ represents the smearing angle due to multiple scattering.

Assuming that individual contributions to the joint probability distribution function of the measurements are Gaussian, the normalized measured residuals squared $R^2(i) = \vec{v}^T(i)S^{-1}\vec{v}(i)$ follow a χ^2_M distribution, where $\vec{v}(i)$ are the measurement residuals, S is their covariance matrix, and M is the number of observations [2]. The recursive expression

$$\lambda(k+1) = \lambda(k) + R^2(k+1) \tag{2}$$

follows a χ^2_{kM} distribution, and is used to determine the goodness of fit of the observations to the helix model.

3.4. Motivations for a Global Approach

Several problems exist with the sequential assignments of vertex detector observations discussed earlier. The nonlinearity of the transport equations induce numerical instabilities due to the application of a nonlinear transformation to the predicted estimators of the state vector. A recursively iterated extended Kalman filter step (i.e., in first-order, a linearization of the filtered estimates obtained by recomputing the Jacobian of the transport equation at the smoothed estimates). has been implemented, but only at great expenses of compute-time, and only with mitigated results [10].

A more serious obstacle is the correlations of observations in terms of track assignment in the outer region, which are not taken into account by a nearest-neighbour assignment rule. Even relatively minor pattern recognition errors there may thus significantly affect the quality of the track estimators in the much more confined vertex detector region, by displacing, or simply enlarging, the extrapolation cone of the outer tracks to cover neighbour observations.

While measurement noise in the vertex detector is nearly Gaussian, the process noise which provides the model for multiple scattering has a Gaussian core but with sensibly heavier tails. The extended Kalman filter estimate thus provides the best linear estimator for the filtered state vector, and in this sense, the procedure is suboptimal. The purpose of the work described in the next section is to reduce the impact of these errors by solving to optimality the combinatorial assignment problem consisting of finding the optimal match of orthogonal pairs of vertex detector observations to outer tracks.

4. Combinatorial Formulation

4.1. The Five-dimensional Assignment Model

4.1.1. VDET Input

As seen in Figure 2, the relatively large gating region of outer tracks cover generally more than a unique combination of vertex detector observations in each layer. The ambiguity contained in this assignment provides the incentive for applying combinatorial optimization to improve the matching and the overall tracking quality.

The intersection of the cones with the wafer planes, together with the observations they cover, define the inputs to a global assignment. In this formulation, a given track may be assigned no observations (when it misses the detector altogether) and up to eight when its extrapolation footprint covers overlap regions in the two layers of the detector. Referring again to Figure 2, the set of input tracks is indexed by i, while the observation candidates in different views and layers are represented by the indices j, k, l and m respectively.

A natural five-dimensional assignment formulation for this problem is achieved by making sure that outer tracks are matched to *exactly one* vertex detector observation per layer and per view. If a track traverse an overlap region, it may effectively be assigned to more than one observation in the same layer and in the same view. We therefore *merge* observations which lie on the overlap region of the detector, in effect increasing the number of logical observations by all possible fused pairs of observations that lie in the same view and the same layer, but on different wafers, which *together* may be assigned to a track.

Noise and inefficiencies of the detector are non-negligible, and contribute to the offset from a perfect 5-dimensional assignment. We introduce therefore a "noise" track with index zero to account for spurious observations. Likewise, "null" observations, in each layer and in each view, are introduced to provide a matching to outer tracks to which no observations can possibly be assigned. In this artificial context, we will refer to observations from now on as "hits", whether real or artificial.

Track assignment put aside, there is no correlation between orthogonal observations which lie on the same layer, so the observations form pairwise-disjoint sets with respect to layers and views. The amplitude of individual observations, however, referred to as their *pulse-height* is measured and allows to determine whether they may be used by one track (labeled from now on "single hits") or more tracks ("undecided hits").

We define the input to our problem as consisting of a set of $(n_i + 1)$ outer tracks, together with $(n_j + 1)$ $\rho - \phi$ hits and $(n_k + 1)$ z hits on the outer layer, and $(n_l + 1)$ $\rho - \phi$ hits and $(n_m + 1)$ z hits on the inner layer. These sets are indexed as follows (see Figure2):

$$i \in \{0, \ldots, n_i\}$$

$$j \in \{0, \ldots, n_{jreal,single}, \ldots, n_{jreal}, \ldots, n_j\}$$

$$\begin{aligned} k &\in \{0, \ldots, n_{kreal,single}, \ldots, n_{kreal}, \ldots, n_k\} \\ l &\in \{0, \ldots, n_{lreal,single}, \ldots, n_{lreal}, \ldots, n_l\} \\ m &\in \{0, \ldots, n_{mreal,single}, \ldots, n_{mreal}, \ldots, n_m\} \end{aligned} \tag{3}$$

If UV_j, UW_k, LV_l and LW_m represent two orthogonal pairs of hits on the outer and lower layers of the vertex detector respectively, the decision variable

$$x_{ijklm} = \begin{cases} 1 & \text{if } T_i \text{ is assigned to } \{ UV_j, UW_k, LV_l, LW_m \} \\ 0 & \text{otherwise} \end{cases}$$

is defined for all outer tracks T_i.

We found it difficult to reduced the number of dimensions in this assignment. A strong correlation between orthogonal hits for a fixed layer is provided by the outer tracks: it would seem unwieldy to assign a single one-dimensional observation to a track in one view, only to find out that no observation is present in the other view. Processing all observations in the layer in one sweep ensures that only meaningful cross-hit patterns are retained. In doing so, however, we run the risk of discarding meaningful patterns which are incomplete because of detector inefficiencies.

To identify bona fide incomplete pattern would in turn require an elaborate parametrization of the local inefficiencies, requiring more precise data reconstruction studies than can possibly be made in the ALEPH context [5]. A workaround to this problem is to extend the search for reasonable patterns into the other layer of the detector. With an expected number of four observations per track, it becomes easier to provide ad-hoc procedures to reject uninteresting pattern candidates at the outset.

Another advantage of processing all vertex detector information at once, as opposed to decomposing the problem up into assignment problems of smaller dimensions, lies in the possibility to exploit the excellent *angular* resolution provided by a *pair* of cross-hits reconstructed in different layers: recalling that an outer track travels a much larger distance in the outer outer region than in the vertex region, the angle of incidence of the tracks with respect to the vertex detector wafers provides a much better assignment criterion with the pair of cross-hits than does its extrapolated position. By keeping the problem five-dimensional, we therefore benefit from this advantage.

4.1.2. Objective function

For all patterns corresponding to the decision variable $x_{ijklm} = 1$, we obtain a a χ^2 value from Equation 2, which we denote by $C_{\chi^2,ijklm}$. A global assignment of these variables (i.e., a solution to the assignment problem) is represented by the vector

$$\vec{X}_{ijklm} = \vec{X}^*_{ijklm} \tag{4}$$

Writing, from now on, the multiple sum of the indices from "a" up to their full range (given in Equation 3) as $\sum_{i,j,k,l,m=a}^{N}$, the number of degrees of freedom for

that assignment, in terms of hits and tracks is expressed as the difference between the total number of real hits and the number of hits assigned to the noise track,

$$\mathrm{NDOF}(\vec{X}^*_{ijklm}) = (n_{jreal} + n_{kreal} + n_{lreal} + n_{mreal}) - \sum_{j,k,l,m=1}^{N} x_{0jklm} \equiv N_{vdet} - N_{unused}$$

The total number of *null* hits for that assignment is

$$N_{null} = \sum_{i=1}^{n_i} \left(\sum_{j=0}^{n_j} x_{ij000} + \sum_{j=0}^{n_k} x_{i0k00} + \sum_{j=0}^{n_l} x_{i00l0} + \sum_{j=0}^{n_m} x_{i000m} \right) \tag{5}$$

To determine whether the hypothesis $\vec{X}_{ijklm} = \vec{X}^*_{ijklm}$ is true, we construct the probability of observation

$$P(\vec{X}_{ijklm} = \vec{X}^*_{ijklm} \mid N_{unused}, N_{null}) = \left(\int_S^{\infty} p_{\chi^2,\mathrm{NDOF}(\vec{X}^*_{ijklm})}(r)dr \right) \tag{6}$$

where

$$S = \sum_{i,j,k,l,m=0}^{N} C_{\chi^2,ijklm} x_{ijklm} \tag{7}$$

Equation 6 is conditional to having observed N_{unused} noise hits, and N_{null} partial patterns given by Equation 5. The probabilistic interpretation of a missing hit can be viewed as a binomial experiment, the outcome of which depends on whether or not the detector records a signal when it should have done so. If $\mathcal{B}$ denotes the binomial probability distribution function (PDF), and ϵ is the local efficiency of the detector on a given wafer chain, the probability of having observed N_{null} partial patterns in the solution $\vec{X}_{ijklm} = \vec{X}^*_{ijklm}$ is given by

$$P(N_{null} \mid \vec{X}_{ijklm} = \vec{X}^*_{ijklm}) = \left(\sum_{I=N_{null}}^{\mathrm{NDOF}(\vec{X}^*_{ijklm})} \mathcal{B}(\epsilon, I, \mathrm{NDOF}(\vec{X}^*_{ijklm})) \right) \tag{8}$$

Likewise, we can treat the occurrence of spurious hits as a Poisson process, whose outcome ranges from zero (no noise is observed) to the total number of observations in the sample (all hits are noise). If λ is the expected number of noise hits in the problem (which is a function of the total number of hits and of the sum of the track extrapolation area), then

$$P(N_{unused} \mid \vec{X}_{ijklm} = \vec{X}^*_{ijklm}) = \left(\sum_{I=N_{unused}}^{\infty} \mathcal{P}(\lambda, I) \right) \tag{9}$$

where $\mathcal{P}$ is the Poisson PDF. The probability of assignment is the product of the last three expressions,

$$P(\vec{X}_{ijklm} = \vec{X}^*_{ijklm}) = P(\vec{X}_{ijklm} = \vec{X}^*_{ijklm} \mid N_{unused}, N_{null}) \cdot P(N_{null} \mid \vec{X}_{ijklm} = \vec{X}^*_{ijklm}) \cdot P(N_{unused} \mid \vec{X}_{ijklm} = \vec{X}^*_{ijklm}) \tag{10}$$

The assignment vector which maximizes expression 10 corresponds to the optimal global assignment of hits to tracks. This expression, however, involves partial gamma functions which are themselves functions of discrete variables, and may not be used directly in an integer optimization formulation. We thus proceed by finding a linear analog to Equation 10, considering the simplified case where the probabilities of the $x_{ijklm} = x^*_{ijklm}$ assignments are all assumed to be independent. Setting a single variable with a nonzero track index to one in the full objective, and taking the negative logarithm of the resulting expression, produces the analog of the conditional probability (now a cost to be minimized) for that pattern

$$c_{\text{cond},ijklm} = -\log\left(\int_{C_{\chi^2,ijklm}}^{\infty} p_{\chi^2,\text{NDOF}_{\text{vdet}}(i)}(t)dt\right) \tag{11}$$

where $\text{NDOF}_{\text{vdet}}(i)$ is the number of hits assigned to i. To this expression, we add a penalty term

$$\gamma_{null,\epsilon} = -\log \mathcal{B}(\epsilon, \text{INT}(4\epsilon), 4) \tag{12}$$

which is weighted by the number of null hits, $N_{null,ijklm}$, present in that pattern (corresponding to the number of j, k, l, m indices equal to zero). Weights given to individual variables x_{ijklm} where i is a real track, are thus

$$c_{ijklm} = c_{\text{cond},ijklm} + \gamma_{null} N_{null,ijklm} \tag{13}$$

We can likewise approximate the contribution of the noise track by

$$c_{0jklm} = -\log \mathcal{P}(\lambda) \tag{14}$$

Equations 13 and 14 define the linear objective function

$$Z(\vec{X}_{ijklm}) = \sum_{i,j,k,l,m=0}^{N} c_{ijklm} x_{ijklm} \tag{15}$$

which is to be minimized.

4.1.3. Constraints

Having introduced null hits in our formulation, real tracks must always be assigned to *some* hit pattern, regardless of its structure. This gives

$$\sum_{j,k,l,m=0}^{N} x_{ijklm} = 1, \quad \forall i \in \{1, \ldots, n_i\} \tag{16}$$

This constraint must be relaxed for the noise track

$$a \leq \sum_{j,k,l,m=0}^{N} x_{(0)jklm} \leq b \tag{17}$$

We merely require that the number of hits assigned to it be bounded by some reasonable number. $0 \leq a$ and $b \leq n_{jreal}+n_{kreal}+n_{lreal}+n_{mreal}$ are two parameters which can be adjusted for this purpose.

Real hit constraints are symmetric (in terms of indices) for each layer and each view, and are thus explicitly written for the $\rho - \phi$ view of the outer layer (indexed by j) only. Other constraints follow naturally by permutation of indices.

For every real hit UV_j, there is an integer $M_j \geq 0$ indicating the number of its occurrences in a fused overlap hit. If that number is non-zero, the vector $G_j(M_j)$ contains the index map of the logical overlap hits (in general, one real hit may belong to more than one overlap pair).

Single hits must be used exactly once. If a hit is part of an overlap combination, mutual exclusion between the hit and its overlap parent in the assignment must be enforced, which yields

$$\sum_{i,k,l,m=0}^{N} \left(x_{ijklm} + \sum_{m=1}^{M_j} x_{iG_j^{(m)}klm} \right) = 1, \forall j \in \{1, \ldots, n_{jreal,single}\} \tag{18}$$

Hits classified as "undecided" are subject to similar constraints. The probability of using these observations more than twice is negligible, given the difference in magnitude between true track separation and vertex detector resolution. This gives

$$1 \leq \sum_{i,k,l,m=0}^{N} \left(x_{ijklm} + \sum_{m=1}^{M_j} x_{iG_j^{(m)}klm} \right) \leq 2, \tag{19}$$
$$\forall j \in \{n_{jreal,single}+1, \ldots, n_{jreal}\}$$

The overlap hits are subject to the constraints which apply to their individual components. No constraints have been placed on the null hits, which are used freely.

4.2. Solution Strategies

4.2.1. Preprocessing

4.2.2. Natural Decomposition

The maximum number of feasible solutions for the five-dimensional assignment model is $(n+1)!^4$, where $n = \max(n_i, n_j, n_k, n_l, n_m)$ is generally dominated by n_i, the number of outer tracks which rarely exceeds 30. Tracks, however, tend to bunch together in subsets which are well separated, so one may successfully apply a partitioning to the main assignment problem in order to produce several logically independent subproblems of smaller sizes. The first preprocessing step consists therefore in checking whether any two tracks in the event belong to the same subcomponent. A simple argument is used in this classification: if we recall that the 5σ road of each track covers a certain number of candidate hit patterns on each layer, we represent all the tracks in the event as the nodes of a graph, in which any two nodes are connected by an edge of positive weight only if the two corresponding tracks share at least one hit amongst their lists of available patterns. The connected components of the graph (identified by a O(n) algorithm using a doubly linked list) then corresponds to so many independent subproblems. Coupled to this step is a verification phase to remove entirely diagonal problems, i.e., instances in which the set of local optimal patterns of each track is in fact the global optimal solution. An example of a physics event involving four logical components is given in Figure 3.

4.2.3. Clustering

A heuristic is now applied to remove from each connected component tracks with an extrapolation error which is much larger with respect to other competing tracks. These tracks may have been poorly fitted in the outer region, or may have undergone large angle multiple scatter. While their χ^2 value provided by the Kalman filter may still be reasonable, either because the fit in the outer region was so poor and only few observations had been assigned to them there, or because their covariance matrix has simply been underestimated, their removal is justified to allow a more balanced competition between near-optimal patterns which belong to more precisely defined tracks. An example of this large footprint discrepancy is given in Figure 4. In a second stage, removed tracks are fitted to observations left unused by the global assignment performed over the component they initially originated from.

Applying proper corrections to the weights of assignment as a function of the extrapolation error could in principle achieve the same goal. However, this approach has the added advantage of reducing the component size, and allowing further decomposition.

The removal is performed by first calculating edge weights for the connected component graph as follows:

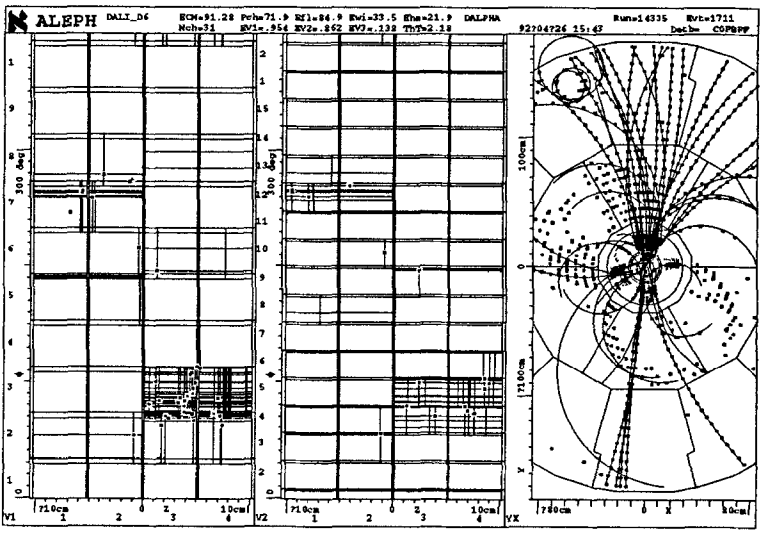

Figure 3. Full event before decomposition: at right a representation of a frontal projection of the outer tracks. Center and left windows represent a normal view of the outer and inner layers of the vertex detector respectively. The leftmost view reveals the presence of four independent components, one of which much more dense than the others (lower part of the picture)

$$w_{ij} = \begin{cases} \frac{(|A_i - A_j|)}{\max(A_i, A_j)} & \text{if } T_i \text{ and} T_j \text{ are connected} \\ 0 & \text{otherwise} \end{cases}$$

where A_i and A_j are the area of intersection between the detector wafer and the extrapolation cones of tracks T_i and T_j respectively, averaged over the two layers. An iteration over the edges of the component is then performed in order of non-decreasing length,and an edge is removed from the graph, together with its predecessors, if its successor is less than half of its magnitude. If the graph has been disconnected in the process, the iterated procedure branches on generated connected subcomponents. An ordering of components is maintained as a function of track index membership. The procedure terminates when all edges have been processed. The assignment is then applied to individual components, in reverse order of their generation.

If $w_{ij}(x)$ is a monotone decreasing function bounded below by $f(x) = \max\{w_{ij}\} - x/2$, where x is the ordering index of the edges, then the clustering step is effectively equivalent to an ordering of tracks as a function of their errors. Applying the assignment over each singleton component, the nearest-neighbour search is recovered.

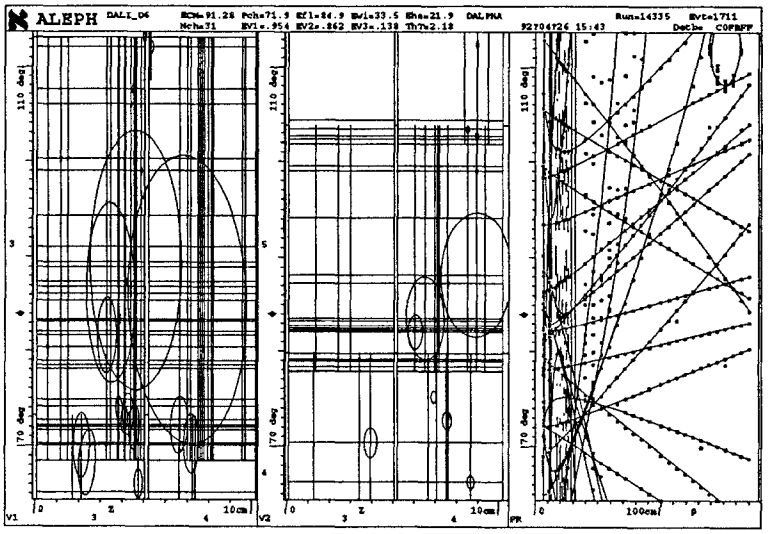

Figure 4. A enlarged view on the dense component shown in the previous figure: two tracks (shown by their very large elliptic footprints at left) correlate all others into forming an oversized component. Removal reduces the component size by half

4.2.4. Branch & Bound

Irreducible subproblems are solved by means of a branch & bound algorithm with linear programming relaxation. This scheme follows the conventional structure of commercial mixed integer programming solver, with some problem-specific steps implemented into the generic structure, among which the use of track momentum for selecting branching variables. If, for a given active node, any two variables with different track indices are in competition, the variable corresponding to the track with the higher momentum, and therefore the smaller extrapolation error, will be branched upon first. Because the footprint of this track is small with respect to others, so is the number of patterns available for it. For variables all involving the same track index, priority is placed on patterns which contain two pairs of real cross-hits (all four hit indices are greater than zero), and, if unavailable, only one pair. Ordering within each subcategory is arbitrary, and the assignment defaults to the null variable X_{i0000} if nothing else is available. The goal of this procedure is to provide an early identification of patterns which are least likely to contribute to the assignment ambiguity.

Node ordering among the list of active nodes follows a depth-first search plus backtracking scheme, with an arbitrary left son selected first (see, for example, Nemhauser & Wolsey [14]).

4.2.5. Post-optimal Processing

Given the suboptimal nature of the procedure described so far, it is only natural to investigate the physical significance of near-optimal solutions, more in an attempt to measure the stability of the optimum itself rather than in providing an analysis of convergence if that optimum is not found. How a small perturbation to the optimal solution may in turn affect the optimal value found earlier is investigated for that purpose. This perturbation is generated by permuting two patterns, one which is part of the optimal solution and one which is not. To render the new assignment feasible requires an interchange of other (and possible all) remaining patterns. Once this has completed, a small difference between the optimal value and the new objective value will indicate the presence of a potentially interesting near-optimal solution, while a large difference will confirm the physical quality of the optimal found earlier.

The selection of an interchange pattern proceeds as follows: for each track, the pattern which is closest to the optimal assignment in terms of absolute weight difference is selected, and the track for which this difference is minimal undergoes the pattern interchange. A list of tracks whose optimal patterns enter in conflict with this new assignment is created, and a greedy search for a feasible solution to this subproblem is performed, which defaults to a null assignment if no pattern may be found for a given track.

5. Implementation

A full implementation of this algorithm has been performed in JULIA, the ALEPH reconstruction software which handles the transformation of raw detector data into fully reconstructed physics events. Although ALEPH is in its sixth year of operations and has already gathered nearly five million Z^0 events, conducting an upgrade at such a late stage in the life of the experiment does not represent a handicap, as raw data, which are stored permanently, are routinely reprocessed. Improvements benefit therefore the entire data sample.

5.1. Selection of Outer Tracks

Reconstruction in the outer region is performed first. This procedure is known to be unstable at times, because the nearest-neighbour search does not handle local correlations of observations with respect to track assignment correctly. Work is in progress to provide a detailed discussion of outer tracking errors [25].

Track selection operates on the basis of a cut on Equation 2, rejecting tracks whose χ^2 value is above an absolute threshold. Also, the track is rejected if its region of extrapolation does not lie in the region of acceptance of the vertex detector. This is true for tracks with their vertex at, or near, the primary vertex, and with an angle of incidence λ greater than 60 degrees.

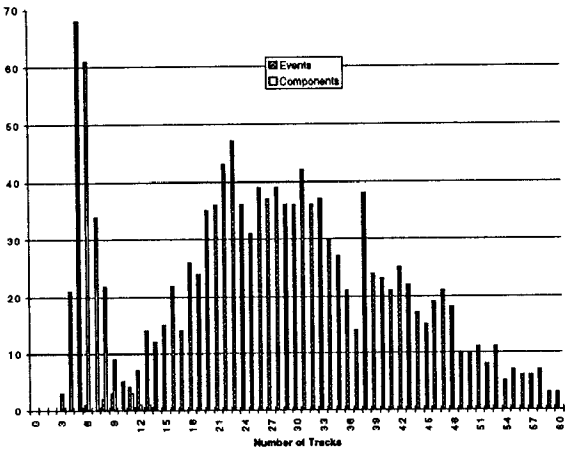

Figure 5. Distribution of the number of tracks per connected component after preprocessing (at left) versus the original distribution of tracks per event

5.2. Problem Size

For a given event, the problem size is essentially determined by the number of tracks and their separation. The center of mass energy of the collision determines the type of particle produced, which, in turn, determine the number of tracks. At the Z^0 energy of 92 GeV, one expects to reconstruct a maximum of around thirty tracks per event, with about twenty one tracks in average. The worst case consists therefore of the assignment of thirty outer tracks to an equal number of observations in the two layers and views of the vertex detector. The corresponding number of feasible solutions is in the order of 10^{128}.

5.3. Computational Experience

A performance study of the procedure was performed on real and simulated data during the testing phase of the implementation. The application of the logical partitioning and clustering steps accounted for a decrease in problem size shown in Figure 5, where the histogram of the number of tracks per connected components is shown after preprocessing (component) and prior to it (event). The mean of the event distribution was about thirty-two (the discrepancy with the expected number of targets is due the same particle being counted more than once if it spirals in the detector). By comparison, the mean was six for the component distribution.

One could expect that the reduction of problem size would come at the expenses of an increase in the *number* of generated subproblems, but two mutually exclusive

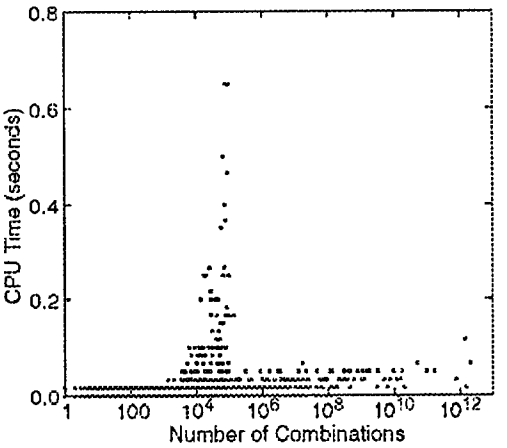

Figure 6. Timing plot of a mixed exhaustive search and branch & bound implementation for small problems, all phases included. The cut-out point between the two methods is clearly visible at 0.68 seconds

outcomes were observed in practice: the number of components in the event was either large but made mostly of diagonal components and included one instance of a dense problem, or the event would partition into three or less subproblems of non-trivial sizes. The original sample of 1134 Z^0 events shown in the figure was in fact *reduced* to 228 components, after removal of diagonal instances, which were solved by sequential search.

As all events, regardless of their size, need to be processed, it seems unwise to apply the same treatment of information to an event with a small number of correlated observations as to a dense event. For very small problems, exhaustive search for the optimal assignment is not only feasible, and also faster than the initialization phase of the branch & bound method. As a practical implementation step, we decided to apply exhaustive search to all subproblems with less than 10^5 possible combinations, and branch & bound to all others. Figure 6 indicates the CPU time per event taken by this hybrid method to solve a sample of a thousand *small* problem instances on a DEC Alpha 3000/300 RISC workstation. The transition in compute-time between exhaustive search and branch & bound, and hence the compute-time gain obtained in applying the global method, is clearly visible at 10^5 combinations (0.68 seconds).

For larger problems, the fully implemented method took in average four times as long as the Kalman-based nearest-neighbour approach. It was observed, however, that calculating the weights accounted for about 90% of the total CPU time, of which 50% was taken by the iterative smoothing step of the Kalman filter. It was observed in later simulation studies that turning off the smoothing step altogether

did not change the efficiency in any but a few events. The filtering step was therefore bypassed, reducing the CPU overhead of the global approach to about 50 % with respect to the original algorithm.

No more than a thousand variables and 150 constraints were ever observed amongst the decomposed instances. The branch & bound algorithm was interfaced with a public domain, dual simplex-based linear programming solver (LP_SOLVE, written by Michel Berkelaar from the Eindhoven University of Technology [3]). All subcomponents processed by this algorithm were solved to optimality, and for that phase alone, compute time never exceeded two seconds of CPU time on a 76 SPECint 92 workstation.

5.4. Simulation Results

To evaluate potential tracking improvements, simulated data were processed by the existing ALEPH reconstruction code, and by the code developed for implementing the global approach. Monte Carlo physics event generators, interfaced with a detector simulation package, provided the simulation platform. Their implementation is discussed in detail in Bonissent [5]. While the generators offer a rather simplistic treatment of data reconstruction errors, and may not be used as a base for detailed analysis, they are useful to for comparative study of this type.

The success rate, in terms of the number of hits correctly assigned with respect to the known simulated solution, was obtained for both methods over a sample of 3000 simulated hadronic events. Considering all events regardless of their complexity, the mean of the success rate distribution was found to be 87.9% and 92.1% for the ALEPH reconstruction code and for the new code respectively. The improvement was more evident when considering only the more dense subproblems, selected by applying a cut of 0.8 on the node/edge ratio, resulting in the distribution shown in Figure 7. In this plot, bin 100 (a 100 % success rate) contains all subproblems for which a perfect match with the Monte Carlo solution was found. The difficulties experienced by the sequential algorithm to handle more complex problem are evident in this figure, as not a single problem could be solved to optimality by this method.

This comparison is incomplete without examining the *failure* rate of the methods, as an observation assigned to the wrong track is more likely to affect the quality of the track than an observations which has merely been removed from it. This is achieved by considering, in the previous sample, the observations which have been assigned to the wrong track by either method, with respect to the Monte Carlo solution, as shown in Figure 8. The combinatorial method is still only slightly better than the sequential assignment. The means of the distributions were 8.3% vs. 6.2% for the two methods respectively. However, considering the results shown in the last two figures, the goal of obtaining a systematic improvement over the sequential method has been achieved.

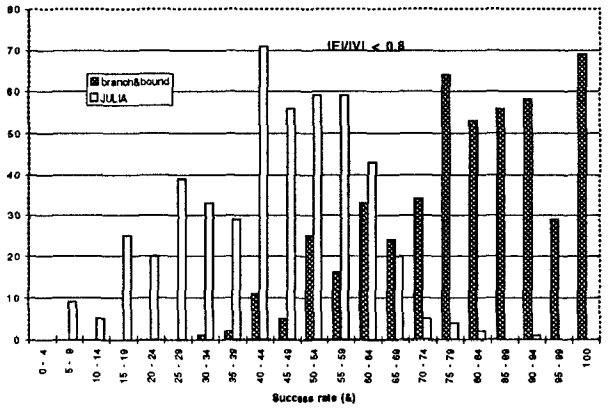

Figure 7. Comparative success rate of JULIA and the branch & bound method, with a cut $\alpha = 0.8$ applied on the edge to node ratio of the component graph

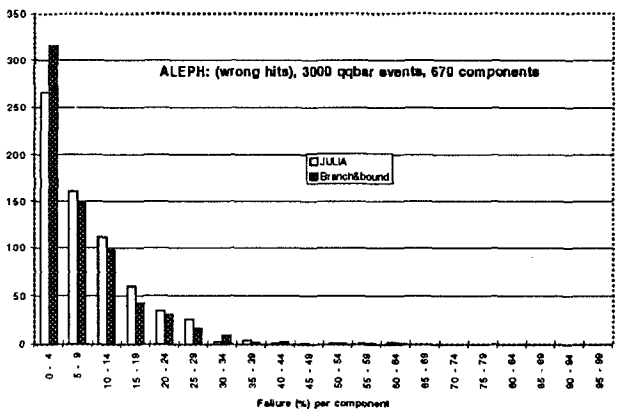

Figure 8. Comparative failure rate of JULIA and branch & bound applied on simulated events. The distribution indicates the number of hits which have *not* been assigned correctly.

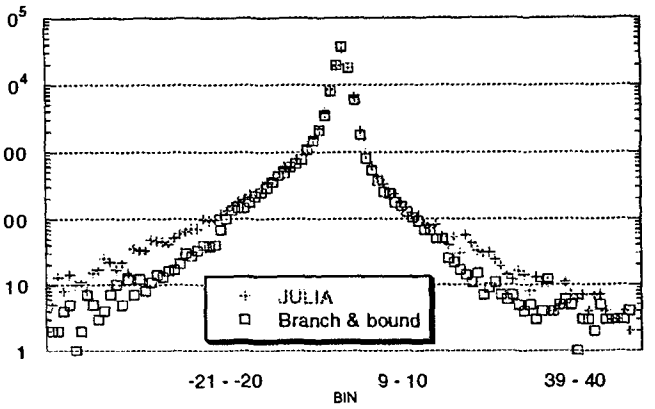

Figure 9. Impact parameter/error distributions for 10'000 hadronic Z^0 events (60'000 tracks), obtained with sequential code (JULIA) and combinatorial method. The branch & bound method described in this paper produces a sharper peak about the impact parameter value (bin zero) and flatter tails in the outlying area

5.5. Physics Analysis with Improved Assignment

It should be noted that the simulation results described earlier clearly reveal that both methods were prone to errors. As it is difficult to isolate the source of these errors as coming from the poor performance of pattern recognition algorithms in the outer tracking, modeling flaws, or simply Monte Carlo errors, this simulation study was not sufficient in itself to justify the usefulness of our tracking approach. To do so, we considered next a comparison of tracking quality over a large sample of *real* data, by solving the so-called "hadron tagging" problem. This consists of identifying in an event the decay signature of a particle which contains a b-quark through the reconstruction of the *impact parameter* of its track, defined as the distance of closest approach between the track and the primary vertex. This result has applications in heavy flavour physics analysis, and is fully described in Brown [6].

The probability that a measured impact parameter is consistent with a hypothesis (e.g., that it comes from the primary vertex) is computed using a resolution function which is measured directly from the data, and which provides a direct measure of tracking quality in the vertex detector.

Figure 9 shows the resolution histogram of the impact parameter divided by its error for 10'000 Z^0 events for both the sequential (JULIA) and branch & bound based methods. The peak of this distribution represents an error-free pattern recognition, while the tail area reveal a poor resolution of the pattern recognition algorithm. The log scale used for this histogram tends to amplify the measurement errors, but

also reveals that the method presented in this paper is also performing consistently better over real data than sequential pattern recognition.

6. Concluding Remarks

The goal of this paper was to show that the data association problem associated with the High Energy Physics tracking problem lends itself well to a hybrid treatment by sequential and global pattern recognition methods. What ultimately determines which method is best is the local degree of correlation of observations with respect to a track formation hypothesis. The vertex detector region in the ALEPH experiment is clearly a region of high correlation, and the combinatorial method proposed here was successfully implemented in its context to provide consistent tracking improvements.

Acknowledgments

We would like to thank L.E. Trotter for many interesting discussions and suggestions provided during the course of this work.

Appendix

The track propagator of the state vector $\vec{r}_k$ at a radius $r_{k+1} = r_k + \Delta r$ is given by [10]:

$$
\begin{aligned}
u_{k+1} &= 2r_{k+1} \arctan \left[\frac{r_k \sin \frac{u_k}{r_k} + \frac{1}{\omega_k} [\cos \Phi_k - \cos (\Phi_k + t_k)]}{r_{k+1} + r_k \cos \frac{u_k}{r_k} - \frac{1}{\omega_k} [\sin \Phi_k - \sin (\Phi_k + t_k)]} \right] \\
z_{k+1} &= z_k + \frac{t_k}{\omega_k} \tan \lambda_k \\
\Phi_{k+1} &= \Phi_k + t_k \\
\lambda_{k+1} &= \lambda_k \\
\omega_{k+1} &= \omega_k
\end{aligned}
$$

where t_k is given by

$$
t_k = 2 \arctan \left\{ \frac{S \cos \alpha}{R - 2T + 2S \sin \alpha} \left[1 - \sqrt{1 - \frac{R \cdot (R - 2T + 2S \sin \alpha)}{S^2 \cos^2 \alpha}} \right] \right\}
$$

and

$$
R = \Delta r \, (2r_k + \Delta r) \qquad S = \frac{2r_k}{\omega_k}
$$

$$
T = \frac{2}{\omega_k^2} \qquad \alpha = \Phi_k - \frac{u_k}{r_k}
$$

References

1. Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993), "Network Flows", Prentice Hall.
2. Bar-Shalom, Y. and Fortman, Th.E.(1988), "Tracking and Data Association", Mathematics in Science and Engineering, Academic Press.
3. Berkelaar, M.R.C.M., LP_SOLVE 2.0, Eindhoven University of Technology, Eindhoven, The Netherlands. Package available via anonymous ftp at ftp.es.ele.tue.nl/pub/lp_solve/.
4. Blackman, S.S. (1986), "Multiple-Target Tracking with Radar Applications", Artech House.
5. Bonissent, A. and Thulasidas, M. (1994), "A New Simulation Program for the Present and Upgraded VDET", ALEPH Note, 94-152, CERN, Geneva.
6. Brown, D. (1992), "Tagging b Hadrons using Track Impact Parameters", ALEPH Note 92-135, CERN, Geneva.
7. Buskulic, D., et al. (ALEPH Collaboration) (1995), "Performances of the ALEPH Detector at LEP", Nuclear Instr. & Methods, A360, 481-506.
8. Cassel, D.G. and Kowalsky, H. (1981), "Pattern Recognition in Layered Track Chambers Using a Tree Algorithm", Nuclear Instr. & Methods, 185, 235-251.
9. Catlin, D.E. (1989), "Estimation, Control, and the Discrete Kalman Filter", Applied Mathematical Sciences 71, Springer-Verlag.
10. Comas, P., Knobloch, J. and Pusztaszeri, J.F., eds. (1996),"ALEPH Event Reconstruction Manual", ALEPH Internal Report, 96-010, CERN, Geneva.
11. Decamp, D. et al. (ALEPH Collaboration) (1990), "ALEPH: A Detector for Electron-Positron Anihilations at LEP", Nuclear Instruments & Methods A294, 121-178.
12. Gyulassy, M. and Harlander, M. (1991), "Elastic Tracking and Neural Networks for Complex Pattern Recognition", Computer Physics Communications, 66, 31-46.
13. Morefield, Ch.L. (1977), "Application of 0-1 Integer Programming to Multitarget Tracking Problems", IEEE Transactions on Automatic Control, vol.AC-22, no.3. 302-312.
14. Nemhauser, G.L. and Wolsey, L.A. (1988), "Integer and Combinatorial Optimization", Wiley.
15. Nemhauser, G.L., et al., eds. (1989), "Handbook of Operations Research and Management Science", vol.1, North-Holland.
16. Papadimitriou, C.H. and Steiglitz, K. (1982), "Combinatorial Optimization: Algorithms and Complexity", Prentice-Hall.
17. Pardalos, P.M., Pitsoulis, L. and Resende, M.G.C. (1995), A Parallel GRASP Implementation for the Quadratic Assignment Problem. In *Solving Irregular Problems in Parallel: State of the Art*, A. Ferreira and J. Rolim, eds., Kluwer Academic Publishers, 111-128.
18. Pardalos, P.M. and Wolkowicz, H., eds. (1994), Quadratic Assignment and related problems. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. Vol. 16, American Mathematical Society.
19. Particle Data Group (1994), "Review of Particle Properties", Physics Review D, Particles and Fields, 50.
20. Peterson, C. and Anderson, J.R. (1987), "A Mean Field Theory Learning Algorithm for Neural Networks", Complex Systems Publications, 66, 31-46.
21. Peterson, C. (1989), "Track Finding with Neural Networks", Nuclear Instruments & Methods, A279, 537-545.
22. Poore, A.B. (1994), "Multidimensional Assignment Formulation of Data Association Problems Arising from Multitarget Tracking and Multisensor Data Fusion", Computational Optimization and Applications, 3, 27-57.
23. Poore, A.B. and Rijavec, N. (1993), "A Lagrangian Relaxation Algorithm for Multidimensional Assignment Problems arising from Multitarget Tracking", SIAM Journal of Optimization, vol.3, no.3, 544-563.
24. Poore, A.B. and Rijavec, N. (1994), "A Numerical Study of Some Data Association Problems Arising in Multitarget Tracking", in *Large Scale Optimization: State of the Art*, W. W. Hager, D. W. Hearn and P. M. Pardalos, editors, Kluwer Academic Publishers B. V., Boston, 339-361.
25. Pusztaszeri, J.F., "Pattern Recognition in a Proportional Wire Drift Chamber", work in progress.

26. Reid, D.B. (1979), "An Algorithm for Tracking Multiple Targets", IEEE Transactions on Automatic Control, vol. AC-24, no.6.
27. Scott, W.T. (1963), "The Theory of Small-Angle Multiple Scattering of Fast Charged Particles", Reviews of Modern Physics, vol.35, no.2, 231-313.
28. Stimpfl-Abele, G. and Garrido, L. (1991), "Fast Track Finding with Neural Networks", Computer Physics Communications, 64, 46-56.