DEUTSCHES ELEKTRONEN-SYNCHROTRON DESY

DESY 80/72
July 1980

PATTERN RECOGNITION IN TRACK CHAMBER SPECTROMETERS

by

H. Kowalski
Deutsches Elektronen-Synchrotron DESY, Hamburg

NOTKESTRASSE 85 · 2 HAMBURG 52

DESY 80/72
July 1980

PATTERN RECOGNITION IN TRACK CHAMBER SPECTROMETERS

H.KOWALSKI

Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany

Abstract:

A new procedure, based on tree-like structures, for finding tracks in multi-layer drift or proportional chambers is presented and discussed. This method has been successfully applied in the track finding programs for the TASSO detector at PETRA. It has proved to be faster and more systematic than the more standard ones in the reconstruction of high energy e^+e^- jet events. Programs of this type should be able to efficiently reconstruct events of even more complicated topologies expected at LEP.

Invited talk given at the International Conference on Experimentation at LEP, 16-20 June 1980, Uppsala, Sweden

Low mass, high efficiency, high resolution and direct digitization are qualities that make multi-layer drift and proportional chambers almost ideal detectors for charged particles in large solid angle high energy physics experiments. Most current or planned storage ring experiments use them as the principal device for tracking charged particles. However, events at high momentum transfers in these experiments typically have collimated jets with high multiplicities and secondary vertices, giving complicated topologies that are challenging for track finding programs. Pattern recognition is further complicated by background hits, and in drift chambers by extra hits due to the left-right ambiguity that arises from the conversion of the drift time to space points. These extra hits are especially difficult to handle since they are inherently near the real tracks. A typical event of this type in the TASSO detector at the DESY storage ring PETRA is shown in Fig. 1. Fast and efficient pattern recognition programs that solve the track finding problem with reasonable computer resources are essential for the utilization of these detectors. Experiments at higher energies, where the jets are expected to be more collimated and have higher multiplicity, are likely to place additional demands on the performance of the programs.

In this talk I will concentrate on the description of a new tree-like algorithm used in the TASSO track finding program [1,2]. This algorithm is faster and it handles complicated jet events in a more systematic way than standard methods do.

In order to show how it differs from more standard track finding methods, we discuss how the more standard road procedure and our new procedure operate in a simple idealized example. Figure 2 illustrates a straight track in a drift chamber, along with the extra hits resulting from the left-right ambiguity. The road method takes any two hits from the innermost and outermost layers and gathers all hits within a road around the track determined by these hits. As shown in figure 3 the first difficulty with this method is that the road generally cannot be too narrow, since the track parameters determined from two hits may be imprecise. Therefore, there is often more than one hit per layer inside the road, and the best track has to be found by fitting all permutations of hits within the road. Fitting is slow, so fitting the large number of possible permutations requires substantial CPU time. The second difficulty results from chamber inefficiencies, which forces the program to cycle through a substantial fraction of all layer pairs to choose the hits that define roads. This means that many of the permutations will be encountered again and again as different layer-pairs are used. These complications diminish the apparent simplicity of the method and make it difficult to understand the performance. For example, it is impossible to verify that all possible track candidates with at most two missing chambers have been found, without carrying the loops over chamber pairs nearly to completion. The question of how many road-defining pairs to use is a cut that must be studied and optimized along with other cuts. Since it is impossible to fit all permutations of hits in all possible roads, programs that use this approach usually include strategies for reducing the number of roads and permutations. This further obscures understanding the performance.

In our new procedure, speed, clarity and control are obtained at the price of working with objects which are more complicated than single hits: the elementary objects of this approach are pairs of hits called "links" (see figure 4) and the "elementary trees" composed of them. Since a link has two hits, a link already determines the parameters of a track in the case of a straight line or a circle coming from the interaction point [1]. The elementary tree of any given link is composed from this link, which is then a trunk of the elementary tree, and all links

---

[1] These are the most important kinds of tracks and the only ones for which this procedure is being used in the TASSO detector.

having the middle hit in common and with approximately the same values of track parameters (which are then branches) of the elementary tree). Of course, a trunk of one elementary tree can at the same time be a branch of another one. Examples of elementary trees are shown in figure 5. A very fast algorithm then connects elementary trees into full trees. The action of this algorithm is illustrated in figure 6. The correct track is recognized immediately as the chain with the maximum number of links within a full tree. In most cases, tracks can be recognized without fitting. Furthermore, the probability that random links build up a full tree is very small. This is because in the process of building up elementary trees, a considerable preselection of links is performed. The construction of full trees from elementary ones leads to a multiple repetition of this preselection.

The above procedure is used in TASSO in two different ways. In the first application all possible links which could belong to real tracks are listed in a systematic way. We distinguish between different types of links; e.g. links between hits of adjacent layers - i.e. one-gap links, links over two or three gaps, links with track parameters in specific ranges etc. At the beginning, the algorithm searches for tracks constructed from one-gap links only, i.e. for tracks without any missing or displaced hits. The next level of search uses one two-gap link together with all one-gap links. In this way tracks are found with one missing or displaced hit. The procedure then continues, allowing more and more many-gap links at each new level, until all tracks are found.

It is possible to arrange this procedure in such a way that the number of constructed trees is kept relatively small and that every track is found only once. After a search at a given level is finished the best track candidates are chosen, their hits are marked and no longer used in the search on subsequent levels. This prevents the same tracks reappearing at the levels requiring fewer hits. For a more detailed description on this procedure see ref. |2|. Note that the property of the algorithm of finding a track only once is not a trivial one. For example, in the standard method the same track is usually found many times at the same level since it can appear within many, only slightly different, roads. The tree procedure is relatively free of biases other than desirable ones, included on purpose. For example, it is naturally biased towards finding track candidates with the maximum number of hits first.

In the second application used for the first step of the TASSO data reduction chain the above procedure is speeded up at the price of giving up some efficiency by working mainly with one-gap links and building up trees in a way similar to the MST algorithm. (see again ref. |2|.)

In both applications the efficiency and the speed of the track finding can be controlled in a straightforward manner by specifying the types of links and the sequence of their use.

One might ask whether the virtues and advantages of the tree methods show up in practical applications. The question is not so easy to answer, since in order to compare different programs, one has to ensure that they work under the same conditions and produce data of the same quality. After studying the performance of programs over long periods of time I have become convinced that tree methods lead to programs which are an order of magnitude faster than the more standard ones. This conlcusion emerged mainly from our internal TASSO experience. In TASSO we developed and used programs of both types. A direct comparison of their performance in event reconstruction under the same conditions showed that the tree programs were a factor 10 to 50 faster than their counter-parts.

Independent of the difference in pattern recognition, the drift chambers of the TASSO and JADE detectors lead to a high track finding efficiency of 97% per track crossing the chamber. These efficiencies were determined with hadronic events from one photon annihilation at 30 GeV. The events of this type are mainly two jet events with an average multiplicity of 12 charged particles, However, due to photon conversion, secondary interactions and backscatters, on average 17 tracks are seen and reconstructed. The track losses are mainly due to secondary interactions, decays and overlap of tracks.

Based on TASSO's experience, I now want to draw some conclusions about the use and possibilities of tree methods for pattern recognition.

In spite of the fact that in TASSO tree methods are only used for the searching for circles from the neighbourhood of the interaction point and for straight lines, the above procedure is not solely restricted to these cases. It can easily be generalized to any type of track and to different detectors, by redefining the meaning of the link and of the elementary tree. For example, for circles not coming from the interaction point, the links have to be built from three hits and elementary trees should be composed of links that have two hits in common. This of course enlarges the lists of links and elementary trees considerably but even larger lists can be efficiently manipulated by modern computers.

It is possible also to generalize the method to detectors that make a dense sequence of measurements on a track. In this case it makes no sense to create links from every pair or triplet of points; rather many points should be gathered in one link. However, care should be taken that the resulting set of links is complete. A complete set of links it is a set from which every possible track can be constructed as a chain of its elements without any interruption. Then the track search can be performed without fits and extrapolations over large regions of the chamber. Tracks are again found as chains (trees) of links using a tree algorithm.

The tree algorithm puts additional requirements on the performance of tracking chambers. The joining of links into elementary trees, which is a procedure of preselecting link combinations, requires good local precision (in contrast to a precision obtained by sampling many measurements). Also a considerable degree of homogeneity of the chamber (e.g. same drift cell size throughout the chamber) is extremely valuable, since the tracks are recognized as chains of only locally related links. Local inhomogeneities (e.g. at the boarders between different chamber types) tend to break the chains and in turn make track finding more difficult.

The tree algorithm is also useful in investigating the pattern recognition properties of a new chamber design before the chamber parameters are fixed. This will allow an optimisation of the number of wires, their positioning etc.

Probably the most important conclusion emerging from TASSO's experience, is that working with the list of all possible track elements and the local relations between them (like links and elementary trees) leads to efficient and fast programs. Programs of this type should certainly be able to reconstruct events of even more complicated topologies as expected from LEP.

Acknowledgements

I would like to thank D.G. Cassel, P. Söding and G. Wolf for criticism and help in preparing this talk.

References:

1) Boerner,H., Fischer,H.M., Hartmann,H., Löhr,B., Wollstadt, M., Cassel, D.G., Kötz, U., Kowalski, H., Wiik, B.H., Fohrmann,R., Schmüser, P., DESY 80/27

2) Cassel, D.G., Kowalski, H., in preparation

Figure Captions

Fig. 1 A typical TASSO one photon annihilation hadronic event
a) without track reconstruction
b) with track reconstruction

Fig. 2 A straight track in the drift chamber along with the extra hits from the left-right ambiguity (the horizontal scale differs from the vertical one)

Fig. 3 A straight track together with the road determined by a hit pair

Fig. 4 A straight track of Fig. 2 drawn as a set of one-gap links. The numbers indicate the addresses of the links in the link list

Fig. 5 List of some elementary trees composed of the links of Fig. 4

Fig. 6 A full tree composed from some links of Fig. 5 and containing a part of a straight track from Fig. 2

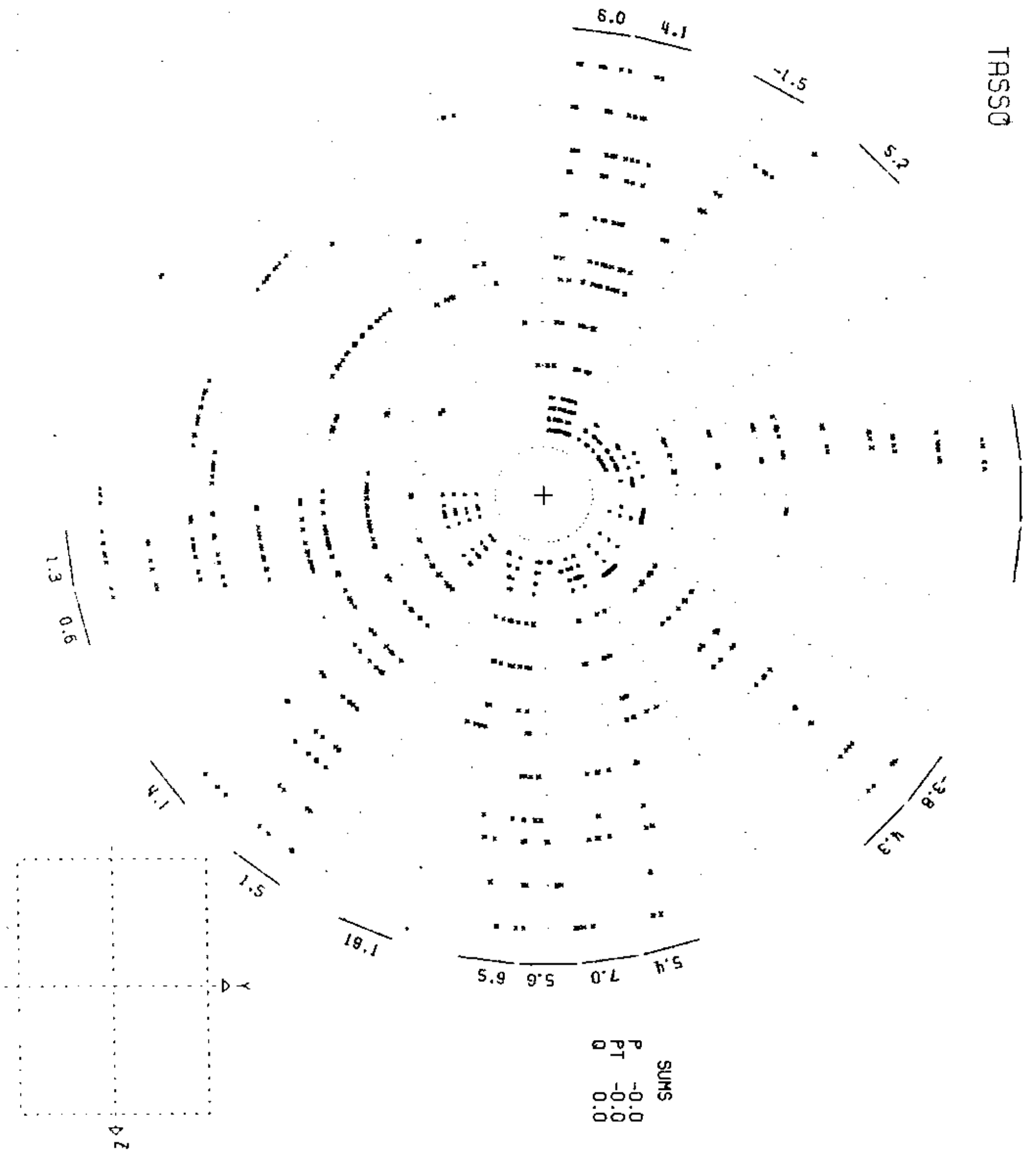


Fig. 1a

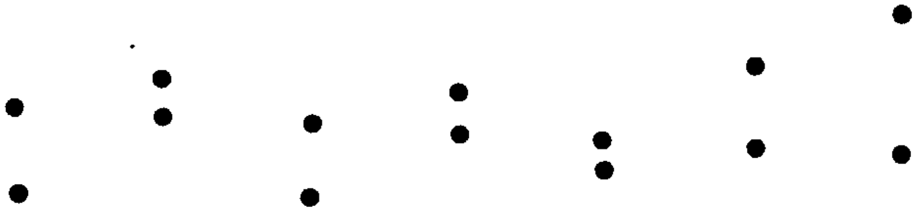
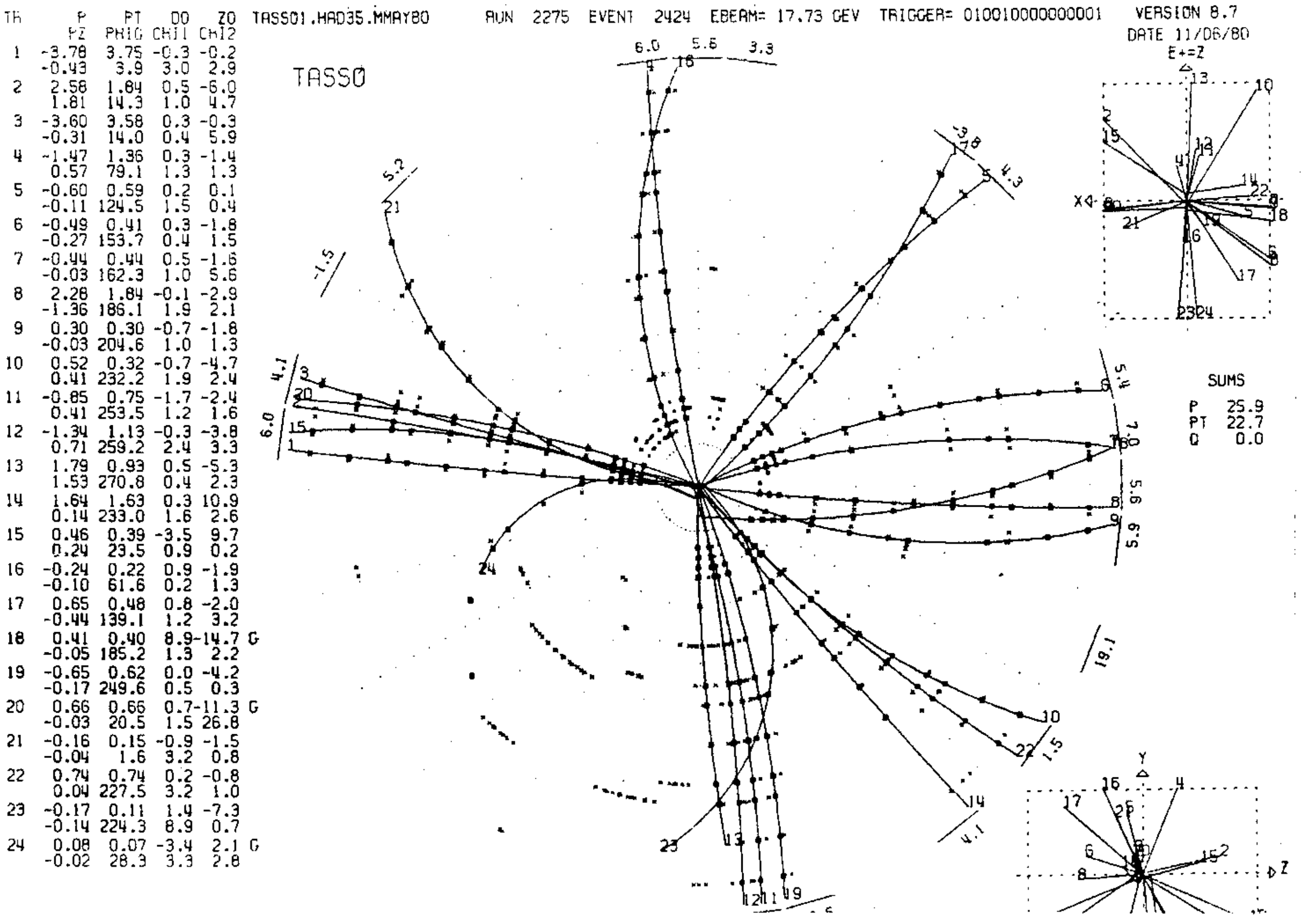


Fig.2



```
TR       P     PT    DO    ZO   TASSO1.HAD35.MMAY80       RUN  2275  EVENT  2424  EBEAM= 17.73 GEV  TRIGGER= 01001000000001    VERSION 8.7
        PZ   PHIG  CHI1  CHI2                                                                                                  DATE 11/06/80
 1   -3.78   3.75  -0.3  -0.2
     -0.43    3.9   3.0   2.9
 2    2.58   1.84   0.5  -6.0
      1.81   14.3   1.0   4.7
 3   -3.60   3.58   0.3  -0.3
     -0.31   14.0   0.4   5.9
 4   -1.47   1.36   0.3  -1.4
      0.57   79.1   1.3   1.3
 5   -0.60   0.59   0.2   0.1
     -0.11  124.5   1.5   0.4
 6   -0.49   0.41   0.3  -1.8
     -0.27  153.7   0.4   1.5
 7   -0.44   0.44   0.5  -1.6
     -0.03  162.3   1.0   5.6
 8    2.28   1.84  -0.1  -2.9
     -1.36  186.1   1.9   2.1
 9    0.30   0.30  -0.7  -1.8
     -0.03  204.6   1.0   1.3
10    0.52   0.32  -0.7  -4.7
      0.41  232.2   1.9   2.4
11   -0.85   0.75  -1.7  -2.4
      0.41  253.5   1.2   1.6
12   -1.34   1.13  -0.3  -3.8
      0.71  259.2   2.4   3.3
13    1.79   0.93   0.5  -5.3
      1.53  270.8   0.4   2.3
14    1.64   1.63   0.3  10.9
      0.14  233.0   1.6   2.6
15    0.46   0.39  -3.5   9.7
      0.24   23.5   0.9   0.2
16   -0.24   0.22   0.9  -1.9
     -0.10   61.6   0.2   1.3
17    0.65   0.48   0.8  -2.0
     -0.44  139.1   1.2   3.2
18    0.41   0.40   8.9-14.7 G
     -0.05  185.2   1.3   2.2
19   -0.65   0.62   0.0  -4.2
     -0.17  249.6   0.5   0.3
20    0.66   0.66   0.7-11.3 G
     -0.03   20.5   1.5  26.8
21   -0.16   0.15  -0.9  -1.5
     -0.04    1.6   3.2   0.8
22    0.74   0.74   0.2  -0.8
      0.04  227.5   3.2   1.0
23   -0.17   0.11   1.4  -7.3
     -0.14  224.3   8.9   0.7
24    0.08   0.07  -3.4   2.1 G
     -0.02   28.3   3.3   2.8

SUMS
P   25.9
PT  22.7
Q    0.0
```

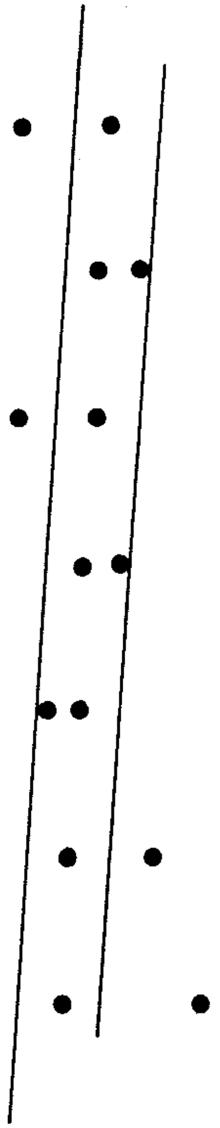


Fig.3

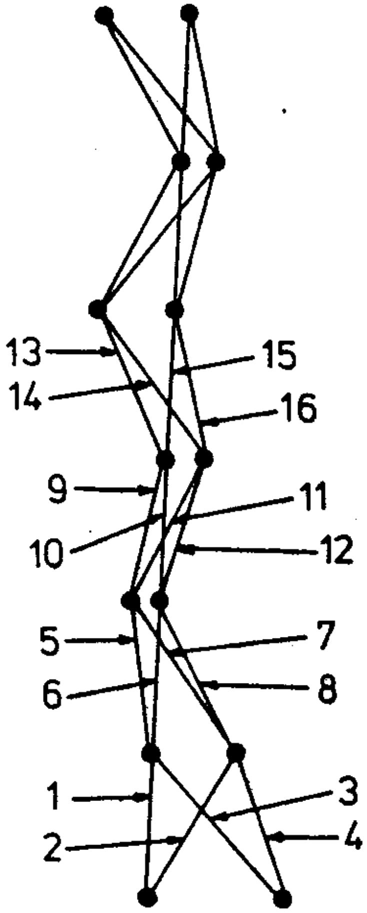


Fig.4

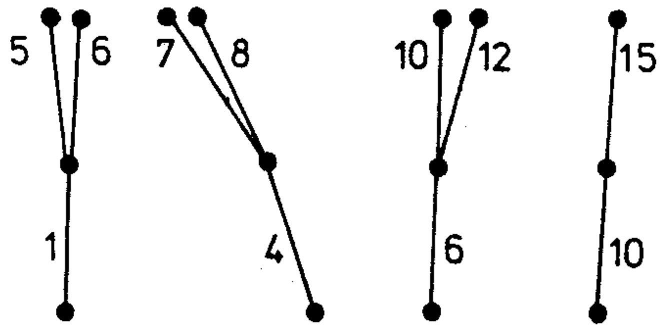


Fig.5

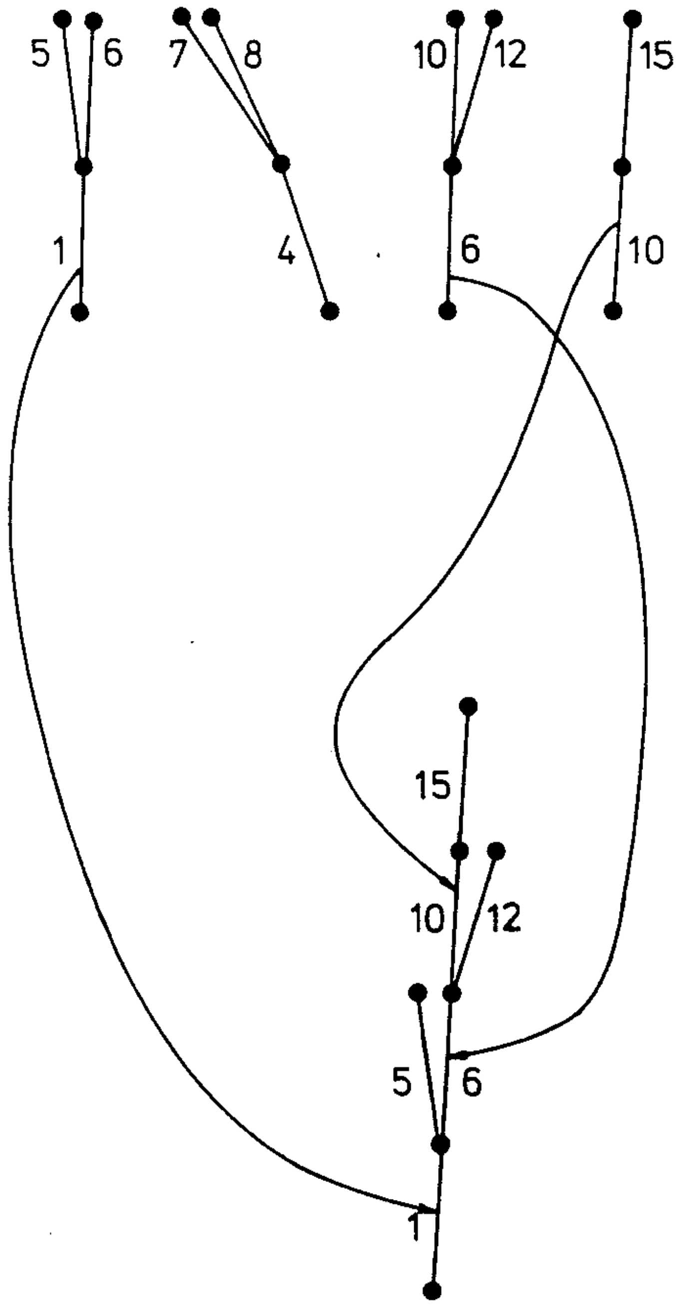


Fig.6