KAM Series:

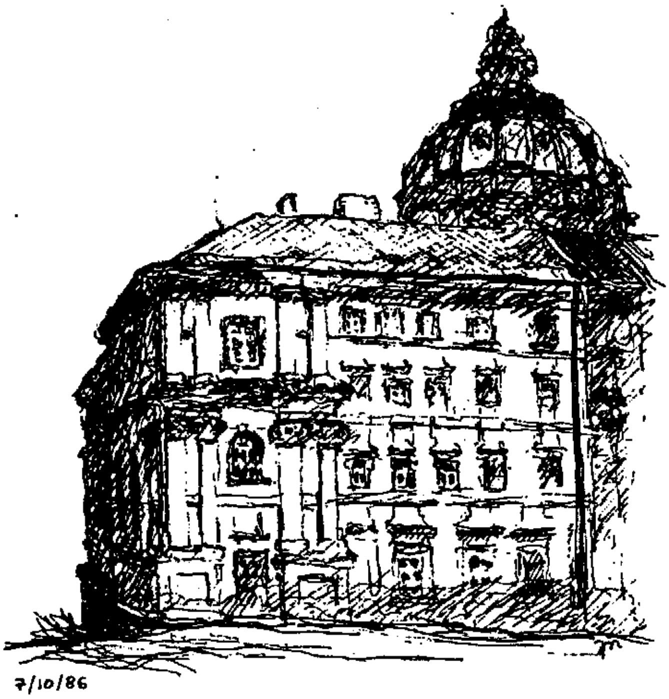
DISCRETE MATHEMATICS AND COMBINATORICS
OPERATIONS RESEARCH
MATHEMATICAL LINGUISTICS

No.
90 - 190

Single linkage vs. complete linkage

M. Křivánek

December 1990



Department of
Applied Mathematics

Faculty of
Mathematics and Physics

Charles University

(KAM MFF UK)
Malostranské nám. 25
118 00 Praha 1
Czechoslovakia

SINGLE LINKAGE vs. COMPLETE LINKAGE

MIRKO KŘIVÁNEK

Department of Computer Science
Charles University
Malostranské nám. 25
118 00 Praha 1, Czechoslovakia

Abstract.

In recent years hierarchical clustering has drawn considerable attention of many researches in various branches of applied mathematics. The aim of this contribution is to provide a practical computer scientist's viewpoint on hierarchical clustering with the emphasis on algorithmical and geometrical aspects. The discussion is devoted to the two well-known methods of hierarchical clustering: single and complete linkages. The underlying question we are focus on is whether the complete linkage admits simple and effective algorithms as the single linkage does. Though leaving this problem open we found some evidences that the complete linkage, at least in the plane, involves more complicated combinatorial structure and algorithms.

1. Introduction

This paper is intended to provide various views on algorithmical and/or geometrical background of single and complete linkage algorithms which are commonly used in hierarchical clustering [JS71, An73, Ja78]. Because of the wide range of everday applications of hierarchical clustering in both social and exact sciences the better understanding of algorithmic structures of hierarchical clustering is desirable. We made first step in this direction by systematically investigating two basic strategies of hierarchical clustering: single and complete linkages. We concentrate on special but important and interesting Euclidean instances, namely plane instances. Our goal is to attract the attention to this amazing area of efficient plane algorithms.

In Section 2. we introduce the single and complete linkage algorithms•and underlying algorithmic structures more precisely. Section 3. is devoted to computational aspects in the plane. We refine the notion of admissibility of [FV71] and show that the complete linkage is not tree-connected admissible, i.e. that it posseses the unpleasant property concerning the 'substantial' overlapping of convex hulls of clusters in the complete linkage hierarchy. This fact complicates the algorithms and causes the increase of the complexity. We also mention the sensitivity to dynamic changes on input data which impacts on the design of dynamic on-line algorithms.

2. Background on single and complete linkages

In recent years a great deal of attention has been paid to hierarchical clustering methods [An73,Ja78]. Two most popular methods which are generally considered as the basis for the design of any algorithmic scheme are called *single* and *complete linkages.*

However, they are also known under various names, e.g. 'max', 'diameter', 'furthest neighbor' refer to complete linkage, whereas 'min', 'connectedness', 'nearest neighbor' refer to single linkage.

The aim of this section is to give a brief summary of what the single and complete linkages do, review some algorithmic paradigms where they belong to and where they represent two opposite poles. We shall also remind some problems encountered with their usage.

2.1 Preliminaries. Throughout this paper let X be a finite set of objects which are to be clustered. Further, suppose that we are given a dissimilarity measure d defined on $X \times X$ that satisfies four constraints :

(1) *symmetry* $d(x,y) = d(y,x), \ x, y \in X,$
(2) *positivity* $d(x,y) \geq 0 \ (\forall x, y \in X),$
(3) *nullity* $d(x,y) = 0 \Leftrightarrow x = y,$
(4) *monotonicity* $d(x,y) \neq d(u,v) \ (\forall x \neq y \neq u \neq v \neq x).$

The goal of hierarchical clustering is to produce a sequence of partitions of X which are nested according to partition refinement, and contains both X and partition of X into singletons. Each partition in the sequence is associated with some integer (or real) value. More precisely, we define a hierarchy $\mathcal{H} = (H, f)$ on X as follows:

(1) $\emptyset \notin H, \ X \in H,$
(2) $(\forall h, h' \in H) \ h \cap h' \in \{\emptyset, h, h'\},$
(3) $(\forall x \in X) \ \{x\} \in H,$
(4) f is a real function defined on 2^X and $(\forall x \in X) \ f(x) = 0,$
(5) $(\forall h, h' \in H) \ h \subseteq h' \Rightarrow 0 \leq f(h) \leq f(h').$

The last property defines so-called *monotone invariant* hierarchies. If the valuation function f is clear from the context we shall not use it explicitely.

The hierarchy is called *binary* if it contains exactly $2n - 1$ clusters. Clearly, we can restrict our investigation (with little loss of generality) to binary hierarchies. It is convenient to represent hierarchical trees graphically by drawing trees (so-called dendrograms), see Figure 1. Here leaves correspond to one-element subsets of X, the root is labeled by X, and each internal vertex is labeled by the union of labels of its 'sons'. Note that each level i in the hierarchical tree defines the *level i* partition of X labeled by the valuation f.

It is well-known that hierarchies are in *one-to-one* correspondence with the set of all *ultrametrics* on X [JS71]. Also computational problems of hierarchical clustering are often formalized as the best fit approximation of given dissimilarity measure by some ultrametric on X. Unfortunately this lead to *NP*-hard optimization problems [KM86]. Hence many approximation schemes of hierarchical clustering are being used in practice. They are discussed in the next subsection.

Final remark of this subsection concerns the possible representation of output hierarchies. We shall prefer the concise representation by so-called *ultrametric path* [Kř 90b]. The ultrametric path P is defined as a weighted path on the set of objects X and it is obtained from the hierarchy $\mathcal{H}$ by taking the ordering of leaves and by 'projecting' values of f onto edges of P, see Figure 1. Note that each edge of P can be associated with corresponding label of the vertex of hierarchical tree, i.e. with the cluster and its corresponding value given by f. In [Kř90b] it is shown that it takes $O(n \log n)$ time to construct the ultrametric path from a hierarchy and vice versa.

2.2 Algorithms. Basicaly two general approximation strategies are recognized in hierarchical clustering: *agglomerative* and *divisive* paradigms. The agglomerative algorithms construct a hierarchy in 'bottom-up' fashion. We start from the partition of X into singletons and at each step we merge two the most similar clusters until we obtain the whole set X. The divisive algorithms start by the whole set X and at each step divide the most 'expensive' cluster into two parts until the singleton partition of X is obtained. The criteria for measuring similarity or cost of clusters distinguish the various methods of agglomerative and divisive hierarchical clustering, see e.g. [Ja78].

Both single and complete linkages are defined with respect to the agglomerative paradigm. Suppose we are given a partition of X. At the current step of single linkage we union two clusters C, C' such that the 'between-cluster'distance $d_{single}(C, C') = \min\{d(x,y) | x \in C', y \in C\}$ is minimized over all pairs of clusters. On the contrary constructing the complete linkage hierarchy we agglomerate that pair of clusters such that the distance $d_{complete}(C, C') = \max\{d(x,y) | x \in C', y \in C\}$ is minimized. The agglomerative paradigm can be easily implemented by means of so-called Lance and Williams recurrence formula [LW67]. Day and Edelsbrunner [DE84] gave $O(n^2 \log n)$ algorithm based on data structure called *heap*. In [Kř90a] the time was improved to optimal $O(n^2)$ by means of so-called $(a,b)-tree$ data structure in amortized complexity sense.

Another implementation of agglomerative paradigm is based on the notion of *symmetric nearest neighbors*, see e.g. [Mu83]. In this case we maintain a directed graph on clusters. Each cluster is joined by a directed edge with cluster which is its nearest neighbor with respect to predefined notion of nearness. A symmetric nearest neigbors are joined by edges oriented in both direction. Note that the agglomeration of the nearest neighbors can be realized by the contraction of corresponding vertices in the graph and that symmetric nearest neighbors can be agglomerated independently of the current agglomeration step. The implementation for both single and complete linkages consumes the optimal $O(n^2)$ time [Mu83].

Finally, it is worth noting that the interpretation of single and complete linkages in graph theoretical terms [Hu74]. As it was several times observed there is a very close connection between minimum spanning trees and single linkage hierarchy [Ro73]. Let us demonstrate this connection in terms of ultrametric paths.

THEOREM 1. *Let T be a minimum spanning tree on X. Then there is an $O(n \log n)$ algorithm that constructs the ultrametric path P from T on X.*

Proof: It is interesting that the underlying algorithm can be implemented either in agglomerative or in divisive fashion. Suppose that the set E of edges of T is ordered. Now, we can proceed either in agglomerative or divisive manner as follows:

```
(A) Agglomerative algorithm
    repeat
      Take out the minimum edge $e = \{x, y\}$ from $E$;
    if $x$ or $y$ $\in$ some component $C$ of ultrametric path $P$ then $C$:=$C\cup e$
    else initialize new component $C = \{e\}$ of $P$
  until 'done'.
(B) Divisive algorithm
  recursive procedure DIVI($T$:  tree):path;
    Let $e$ be the maximum edge of $T$;
    if $T$ has exactly one edge then $P$:=$T$ and return
    else divide $T$ into $T_1$ and $T_2$ by cutting $e$;
       $P$:= DIVI($T_1$) $\cup$ DIVI($T_2$); return.
```

The agglomerative algorithm repeatedly constructs pieces (components) of an ultrametric path P on X. Note that it uses so-called disjoint union-find set algorithm [Ta83]. Thereby its time complexity is $O(n\alpha(n))$, where α is the extremely slowly growing functional inverse of Ackermann function [Ta83]. On the other hand the recursive implementation of divisive algorithm constructs the ultrametric path by successive refinements. In fact it also relies on disjoint union-find mechanism and has the same time complexity.

However, the preprocessing sorting step requires $O(n \log n)$ steps and thus it dominates the time complexity. □

Let us remind that the time needed for the construction of minimum spanning tree in the complete graph is $O(n^2)$ [Ta83]. The size of the input instance is of the same order and hence the ultrametric path can be constructed in optimal quadratic time.

On the other hand the complete linkage is related to graph coloring problem on complements of so-called threshold graphs on X [BH76]. A threshold graph G_t is a subgraph of the complete graph on X that it has edges with weights less or equal to the threshold t. Unfortunately, the relationship does not imply an efficient algorithm for the complete linkage (graph coloring is NP-hard [GJ79]) and the complete linkage can be thought of to be only an appoximation scheme for graph coloring.

2.3 Euclidean instances. In consistency with 'practical' clustering we focus our investigations on Euclidean input instances. The special attention will paid to the plane instances. We shall assume that objects are points in coordinate d-dimensional Euclidean space E^d and that the dissimilarity d is given by the ordinary Euclidean metric ρ. It is worth noting that standard methods of factor analysis, e.g. the method of principal components, are available for embedding higher dimensional instances into the plane more or less faithfully. However, in general isometric embeddings are not possible [Sch38] and therefore some approximate solution are desirable, let us mention interesting papers of [LSB77] and [Ma90].

Now, let us discuss the particulars of Euclidean instances from the computational viewpoint. First of all the size of an input instance is of order $O(n)$ (in fixed dimension) since all distances are implied and they are computable in constant time by well-known formula. Further, we can consider, as in the general case, that the monotonicity constraint is also valid. Moreover, we may assume that points are in general positions (no three points are colinear, no four points are cocircular ...) since several pertubation methods can be applied in preprocessing [Ro90]. Finally, we describe briefly the general 'tie-breaking' convention of [MPSY88] which guarantees mutual distinctness of distances in the plane. The 'new' set of objects is produced by shifting each point right by a very small distance. This distance can be chosen so small that does not affect the relative ordering of any distance. We can proceed as follows. We order objects in X lexicographically by their coordinates. Then we consider for each pair $x_i, x_j, (i < j)$ the triple $< \rho(x_i, x_j), j, -i >$. Lexicographic ordering on these triples, when projected onto the first coordinate, gives the required ordering of distances.

3. Computational aspects in the plane

In this section we study single and complete linkages in the plane from various viewpoints. Since the lower bound complexity of agglomerative hierarchical clustering in the plane is of order $O(n \log n)$ [Kř90b] our aim is to develop optimal algorithms having the same complexity.

3.1 Connected admissibility. Fisher and van Ness [FV71] introduced general admissibility conditions which are desirable under almost any circumstances in hierarchical clustering analysis. First, we shall deal with a slightly generalized notion of connected admissibility.

We shall say that hierarchical clustering procedure is *connected admissible* if it produces a hierarchy H on X such that

$$(\forall h, h' \in H), h \cap h' = \emptyset \;\Rightarrow\; \mathrm{mst}(h) \cap \mathrm{mst}(h') = \emptyset,$$

where $\mathrm{mst}(h)$ denotes the minimum spanning tree on vertices of h.

Note that convex hulls $CH(h), CH(h')$ need not be disjoint. Let us remind that the connected admissible hierarchy with the additional convex hull disjointness requirement is called *convex* admissible [FV71]. It is known that single linkage is connected admissible while complete linkage is not connected admissible [FV71]. Overlapping clusters of complete linkage hierarchy seem to be disadvantageous. Actually, we are able to localize 'the extent of overlapping' and show that the most disagreeable case ocurrs with the complete linkage hierarchy. For this sake we define the notion of *tree-connected* admissibility by the following condition, c.f. Figure 2:

$$\begin{aligned}(\forall h, h' \in H), h \cap h' = \emptyset \;\Rightarrow\; &\exists \text{ spanning tree } T \text{ on } h \text{ and spanning tree } T' \text{ on } h' \\ &\text{such that edges of } T \cap \text{ edges of } T' = \emptyset.\end{aligned}$$

Let us consult Figure 2 where case (a) illustrates the notion of tree-connected admissibility and the case (b) demonstrates the hierarchy which is not tree-connected admissible. Hence the tree-connected admissibility admits reasonable overlapping of clusters.

LEMMA 1. *A hierarchy $\mathcal{H}$ is tree-connected admissible if and only if*

$$\forall h, h' \in H, h \cap h' = \emptyset \;\Rightarrow\; |CH(h) \cap CH(h')| \leq 2.$$

Proof: Obvious. □

Further, we have another trivial lemma.

LEMMA 2. *The single linkage hierarchy is tree-connected admissible.* □

On the contrary we cân prove the following.

THEOREM 2. *The complete linkage hierarchy is not tree-connected admissible.*

Proof: The proof is illustrated by an on-scale construction in Figure 3. □

Let us remark that Theorem 2 characterizes the so-called migration effect which is treated in more detail in the next subsection. There is also the linear algorithm which transfers the not tree-connected admissible hierarchy into the convex admissible hierarchy. The algorithm is also mentioned in Section 3.3.

3.2 Dynamic point sensitivity and migration effect. The aim of this subsection is to discuss the influence of outlying points ('noise') to the behaviour of both single and complete linkages. We also mention the migration effect.

According to [FV71] we define the cluster omission admissibility as follows. The hierarchical clustering procedure is called *cluster omission* admissible if it constructs a hierarchy H on X if and only if it constructs for each $h \in H$ a hierarchy $H' = \{h' - h | \ h' \supseteq h\}$ on $X - h$. In [FV71] it is stated that both single and complete linkages are cluster omission admissible. Let us look more carefully to this concept. Let us think about the hierarchy H as about a tree. If we now delete a cluster $h \in H$ it means that the new hierarchy on $X - h$ is obtained by removing h from the tree and by restructualizing the tree starting from the clusters on the same level as h had. However, in the case of the plane single linkage hierarchy we may hope for better result. Note that due to the relationship of minimum spanning trees with the single linkage hierarchies, the cluster h is associated (via the valuation f) with the minimum edge of the cut between $h', h'' \in H$ such that $h' \cup h'' = h$. This minimum cut edge is the edge of a minimum spanning tree on X. Hence the worst case of reconstruction of the hierarchy occurs when the cluster h is the nearest neighbor of all clusters of the same level in H. In another words the hierarchy H was built up by successively join of remaining clusters to h. However in the Euclidean space the maximal number of clusters (points) for which h is its nearest neighbor is $O(1)$, and in the plane is exactly six, cf.e.g.[DE84]. We have the following.

THEOREM 3. *The restructualization of the plane single linkage hierarchy after the cluster omission (deletion) takes constant time regarding the revaluation of f and time proportional to the height of the hierarchy in order to delete omitted cluster from all superclusters.* □

Now, let us turn our attention to the plane complete linkage hierarchy. In this case the restructuralization of a hierarchy seems to be more complicated. Let us pose the question how many clusters can be influenced in the worst case by deleting of one cluster from the plane complete linkage hierarchy. The answer is $O(n)$ - consider for example n clusters as a bunch of non-parallel rays emanating from some circular neighborhood of the origin and one cluster lying in the origin. Then the deletion of the last cluster causes the complete restructuralization of the hierarchy. However, the latter partition, which is by the way quite natural with the single linkage hierarchy, seems hardly to occur in the context with the plane complete linkage hierarchy. We conclude as follows:

THEOREM 4. *The time needed for the restructuralization of the (plane) complete linkage hierarchy is proportional to the height of the hierarchy.* □

The inverse operation to cluster deletion is cluster insertion. More precisely the problem is how to merge together two hierarchies. This type of question arises with the use of divide and conquer strategies in algorithm design. In the case of single and complete linkages the problem is translated to the merge of two ultrametric paths. It is not hard to find examples where the merge step involves the construction of the completely new ultrametric path (even in the plane). Of course some particular cases are of the interest. For example when we want to insert relatively compact cluster or if we merge to sufficiently distant hierarchies. The most interesting question is about point insertions and we discuss it in detail in the subsection of on-line algorithms. It is worth to mention the problem of so-called *migration effect* at this place. With the connection of single linkage the migration effect is presented as so-called *chaining effect* [JS71]. It consists in the fact that two clusters may be agglomerated even in the case when the diameter of their union is large. In the plane case such example is given by points distributed along the line. Notice that the height of resulting hierarchy is $O(n)$. The migration effect with respect to complete linkage is mentioned in the Section 3.1. It may result in not tree-connected admissibility. However, it can be observed that the complete linkage leads to hierarchies such that their structure is similar to full binary trees, cf.[HS75], and thus the height is of order $O(\log n)$.

3.3 Optimality criteria. With the practical usage of the single and complete linkages one may ask about the objective function of these strategies. Since hierarchies on X are in one-to-one correspondence with ultrametrics on X the computational problem of hierarchical clustering is usually expressed as the approximation of a given dissimilarity measure by some ultrametric on X. Here the role played by the valuation f is important since it express the 'quality' (tree structure) quantitatively by unambiguously inducing the ultrametric on X. Unfortunately the underlying decision problems are mostly NP-complete [KM86]. In this respect single and complete linkages serve as the polynomial approximations. First we shall discuss the objective of single and complete linkages in terms of ultrametrics.

Let u be an ultrametrics on X. We shall say that the ultrametric u is *maximal subdominant* to dissimilarity measure d (or ρ), $u \prec d$ if $\forall x, y \in X, u(x,y) \leq d(x,y)$ and there is no ultrametric u' on X such that $u \prec u' \prec d$. Similarly we define *minimal dominant* ultrametric to d. It is easy to observe that single linkage hierarchy induces maximal subdominant ultrametric while clearly the complete linkage induces dominant ultrametric which need not be minimal. The problem to find minimal dominant ultrametric for given dissimilarity measure is NP-hard [Kř88].

Since the hierarchical clustering provides the sequence of partitions of X it is also used for finding the partition (clustering) of X which reflects compact well-separated clusters. It stimulates the investigation of single and complete linkages with respect to optimality criteria placed locally on level partitions rather than globally on hierarchies.

The quality of the clustering (partition of X) is usually expressed by means of *separability* and *closeness*. The separability is measured by various statistics defined on *cuts*, i.e. on the set of all between-clusters dissimilarities. On the other hand the closeness is measured by some statistics taking into account dissimilarities within clusters.

Let Φ be a functional defined on 2^X and expressing the quality of partitions. Let H be a hierarchy on X. We shall consider n level partitions $P_0, \ldots, P_{n-1}$ induced by H.

We shall say that a hierarchy H is *threshold* admissible if there exists a functional Φ such that

$$\forall i\ \Phi(P_i) \text{ attains its minimum over all partitions of } X \text{ into } n-i \text{ classes.}$$

Though the threshold admissibility is quite natural property it seems to be a very strong requirement for hierarchal clustering algorithms. Practically it means that, when constructing the hierarchy, we are not dependent on the agglomeration process. In other words we can proceed in so-called *greedy* way. For example we can construct a Φ-optimal partition of X and then continue by agglomerations and divisions. It turns out that the single linkage nicely illustrates this concept.

As usual we shall use the tight connection of single linkage hierarchy with minimum spanning trees in order to show that single linkage is threshold admissible.

THEOREM. *The single linkage hierarchy is threshold admissible.*

Proof: We use the following criterion Φ related to level i of a hierarchy:
"min cut edge is maximized among all partitions of X into $n-i$ clusters." □

Let us remark that this property enables us to construct single linkage hierarchy following the divisive paradigm. Again we can use minimum spanning tree and recursively delete its maximum edge, cf. also Theorem 1.

Now, let us turn our attention to the complete linkage. One may speculate on complete linkage to be the dual counterpart to the single linkage and formally replace 'max-min' criterion by 'max-max' criterion, or in another words to relate complete linkage hierarchy to maximum spanning tree on X. The reader should have little difficulty in finding an example contradicting to such a conjecture. Moreover, the divisive implementation of max-max criterion of the complete linkage strategy is completely ambiguous. In addition, it is not guaranteed that it results in the complete (agglomerative) linkage hierarchy. Indeed, in divisive strategy we choose the maximal edge of the instance graph and of course there are $O(2^n)$ possibilities of bi-cuts that contain this edge. Since, as we already mentioned, there is a tendency of complete linkage to produce full binary hierarchies we should restrict the search to balanced cuts. But even in that case there are exponentially many possibilities of the choice and moreover related problems on balanced cuts are known to be intractable [GJ79]. The fail of this reasoning about the complete linkage stems in the observation that in the complete linkage we maintain the diameters of the union of two clusters rather than cuts. In other words, the single linkage produces some kind of well-separated hierarchies whereas the complete linkage well-compact hierarchies.

The natural choice of the functional Φ for the complete linkage is the following:
"minimize the maximum diameter of clusters among all partitions of X into $n-i$ classes."

Unfortunately, the attemps to establish treshold admissibility of the complete linkage via this criterion fails. This justifies the observation that the complete linkage is inherently agglomerative:

THEOREM. *The complete linkage is not (diameter) threshold admissible.* □

We close the subsection by noting that there exists the algorithm which transfers the given partition of the plane set X such that the convex hulls of its clusters overlap into new partition such that the maximum diameter of clusters of the new partition is not increased and the convex hulls of its clusters are mutually disjoint. The algorithm is a modification of the technique of [CRW90] and is analyzed in [Kř90b].

3.4 Plane algorithms and underlying geometric structures. In the plane we might hope for faster algorithms running in optimal $O(n \log n)$ time. Again, the single linkage confirms our wishes. The algorithmic efficiency is obtained via the connection to the minimum spanning trees. Let us recall that minimum spanning tree in the plane can be found in $O(n \log n)$ time [PS86]. The underlying important structure is so-called *Voronoi diagram*, see Figure 4a. Recall that each point from X is associated with the convex region of plane points that are closer to this point than to remaining points from X and that dual graph of Voronoi diagram is a superset of the minimum spanning tree on X. As usual in the context of complete linkage the finding of the corresponding geometric structure is getting more difficult. The framework of special Voronoi diagram is described in [EGS89], see Figure 4b. Here the Voronoi diagram is defined as the partition of the plane into not necessarily connected regions, one for each cluster, such that plane points lying in the region associated with some cluster from X are closer to this cluster than to remaining clusters. The closeness is measured by diameters. It is shown that the combinatorial complexity (number of vertices, edges and faces) of such Voronoi diagram is of order at least $O(n^2)$ for clusters whose convex hulls are not linearly separable. In the case of linear separability the complexity of diagram reduces to $O(n)$. This is another evidence that complete linkage should be modified in order to produce tree-connected admissible hierarchies.

Finally, we shall mention plane strategy based on symmetric nearest neighbors. In the case of the single linkage we can construct so-called all *nearest neighbor graph* by means of Voronoi diagram in $O(n \log n)$ time. Since the maximum in-degree is 6 we can maintain the graph in additional $O(n)$ time. By the contrary the maximum degree of the all nearest graph for the complete linkage can be $O(n)$ and it implies the overall $O(n^2)$ time complexity for the complete linkage based on symmetric nearest neighbors.

3.5 On-line plane algorithms. The aim of this subsection is to provide an introduction to on-line algorithms for both single and complete linkages. It seems to be the promissing area for further investigations and practical efficient algorithmization. On-line algorithms are connected with the concept of *dynamization.* We assume that points that are to be clustered are obtained successively and the clustering structures (hierarchies) are modified by insertions. We also mention the problem of deletions and changes of dissimilarity.

The problems of on-line algorithms for single linkage are fully covered by the results of [EITTWY90]. Here it is shown that even in the weighted planar graph the following operations can be performed in $O(\log n)$ time: changes in edge weights, insertions and/or deletions of vertices and edges. This result is important since it can be used with procedures based on reweighting of minimum spanning trees or Delaunay triangulations in connected admissible hierarchical clustering [Kř90a].

As far as the complete linkage is concerned the first $O(n^2)$ algorithm for the complete linkage [De77] is incremental. This algorithm can be used in the framework of ultrametric paths, too [Kř90b]. The kernel of a current incremental step consists in the following. Let $\mathcal{H}_i$ is the complete linkage hierarchy on $\{x_1, \ldots, x_i\}$. How the hierarchy $\mathcal{H}_{i+1}$ on $\{x_1, \ldots, x_i, x_{i+1}\}$ can be obtained from $\mathcal{H}_i$?. We are to find $h \in \mathcal{H}_i$ such that the distance from x_{i+1} to h is minimal and not grater than $f(h'), h' \supset h$. Hence we must localize the relative ordering of x_{i+1} on the corresponding ultrametric path. Then the edges of ultrametric path have to be reweighted in order to reflect the insertion of x_{i+1}. Note that the reweighting is related to the height of the tree above the ultrametric path and that it may cause the change in the shape (structure) of the tree. Unfortunately in the worst case we cannot improve this idea upon the overall quadratic time.

On the other hand in the case of tree-connected admissible hierarchy we are able to transform the problem to dynamic planar location problem which can be maintained in dynamic $O(\log^2 n)$ time [PT89] per operation.

4. Conclusion

We have shown that the complete linkage involves more complicated algorithms and (geometric) data structures than those that are available for the single linkage. In spite of the fact that single and complete linkage strategies are lying on extreme poles of various algorithmical strategies for hierarchical clustering the complete linkage cannot be handled by greedy-like algorithm. The main algorithmical drawback of the complete linkage is that it can produce not tree-connected admissible hierarchies. It is exactly what causes quadratic complexity of underlying geometric and algorithmic structures in tyhe case of plane instances.

We left open the challenging problem of the construction of the subquadratic, maybe $O(n \log n)$ algorithm for the complete linkage. The promissing approach seems to be the work on algorithm running in $O(n \log n)$ average time. We are now working on such algorithms based on so-called random sampling technique [CS89]. However the better understanding to the complete linkage is still needed, e.g. the probability anlysis similar to that of [Li73] made for the single linkage.

From the practical point of view we advocate the approximations by means of generalized single linkage algorithms [Kř90a] on modified weighted Delaunay triangulations that faithfully preserve distances of the inputs [LL89].

References

[An73] ANDERBERG M.: Cluster analysis for applications. Academic Press, 1973.

[BH76] BAKER F., HUBERT L.: A graph-theoretic approach to goodness of fit in complete-link hierarchical clustering, JASA 71(1976), 870–878.

[CS89] CLARKSON K., SHOR P.: Application of random sampling in computational geometry, II, Discr. and Comput. geometry 4(1989), 387–421.

[CRW90] CAPOYLEAS V., ROTE G., WOEGINGER G.: Geometric clusterings, to appear in Journal of Algorithms, 1990.

[De77] DEFAYS D.: An efficient algorithm for a complete link method, Comput.Journal 20(1977), 364–366.

[DE84] DAY W.H.E., EDELSBRUNNER H.: Efficient algorithms for agglomerative hierarchical clustering methods. Journal of Classification 1(1984), 7–24.

[EITTWY90] EPPSTEIN D., ITALIANO G.F., TAMASSIA R., TARJAN R.E., WESTBROOK J., YUNG M.: Maintanance of a minimum spanning forest in a dynamic planar graph, Proc. 1st ACM-SIAM Symp. on Discrete Algorithms, 1990, 1–11.

[EGS89] EDELSBRUNNER H., GUIBAS L.J., SHARIR M.: The upper envelope of piecewise linear functions: algorithms and applications, Discrete and Comput. Geom 4(1989), 311–336.

[FV71] FISCHER L., van NESS J.W.: Admissible clustering procedures, Biometrika 58(1971), 91–104.

[GJ79] GAREY M., JOHNSON D.: Computers and intractability: a guide to the theory of NP-completeness. Freeman, San Francisco, 1979.

[HU74] HUBERT L.: Some applications of graph theory to clustering, Psychometrika 39(1974), 283–309.

[HS75] HUBERT L., SCHULZ J.: Hierarchical clustering and the concept of space distortion, British J.of Mat. and Stat. Psychology, 28(1975), 87–111.

[Ja78] JAMBU M.: Classification automatique pour l'analyse des donneés. Dunod, Paris, 1978.

[JS71] JARDINE N., SIBSON R.: Numerical Taxonomy. Wiley, 1971.

[Kř88] KŘIVÁNEK M.: The complexity of ultrametric partitions on graphs, Inf. proc. letters 27(1988), 265–270.

[Kř90a] KŘIVÁNEK M.: Connected admissible hierarchical clustering, submitted, 1990.

[Kř90b] KŘIVÁNEK M.: Algorithmic and geometric aspects of cluster analysis. Monograph, to be published by Academia, Prague, 1990.

[KM86] KŘIVÁNEK M., MORÁVEK J.: NP-hard problems in hierarchical-tree clustering, Acta Informatica 23(1986), 311–323.

[Li73] LING R.F.: A probability theory of cluster analysis, JASA 68(1973), 159–164.

[LL89] LEVCOPULOS C., LINGAS A.: There are planar graphs almost as good as complete graphs and as short as minimum spanning trees, Springer LNCS 401, 1989, 9–13.

[LSB77] LEE R.C.T., SLAGLE J.R., BLUM H.: A triangulation method for the sequential mapping of points from N-space to two-space, IEEE Trans. on Computers, C-26(1977), 288–292.

[LW67] LANCE G., WILLIAMS W.: A general theory of classificatory sorting strategies. 1. Hierarchical systems, Computer J., 9(1967), 373–380.

[Ma90] MATOUŠEK J.: Bi-Lipschitz embeddings into low-dimensional Euclidean spaces, Tech.rep.No. 77, Department of Computer Science, Charles University, Prague, 1990.

[Mu83] MURTAGH F.: A survey of recent advances in hierarchical clustering algorithms, Computer J. 26(1983), 354-359.

[MPSY88] MONMA C., PATERSON M., SURI S., YAO F.F.: Computing Euclidean maximum spanning trees, Proc. 4th ACM Symp. on Computational geometry, 241–251.

[PS86] PREPARATA F., SHAMOS M.I.: Computational geometry - an introduction. Springer, 1986.

[PT89] PREPARATA F., TAMASSIA R.: Fully dynamic point location in a monotone subdivision, SIAM J.Comput., 18(1989), 811–830.

[Ro73] ROHLF F.J.: Algorithm 76. Hierarchical clustering using the minimum spanning tree, Computer J. 16(1973), 93–95.

[Ro90] ROSENBERGER H: Degeneracy control in geometric programs, Tech.rep. No. UIUCDCS-R-90-1601, Department of Computer Science, University of Illinois at Urbana-Champaign, 1990.

[Sch38] SCHOENBERG I.J.: Metric spaces and positive definite functions. Trans. AMS 44(1938), 522–536.

[Ta83] TARJAN R.E.: Data structures and network algorithms. SIAM, 1983.

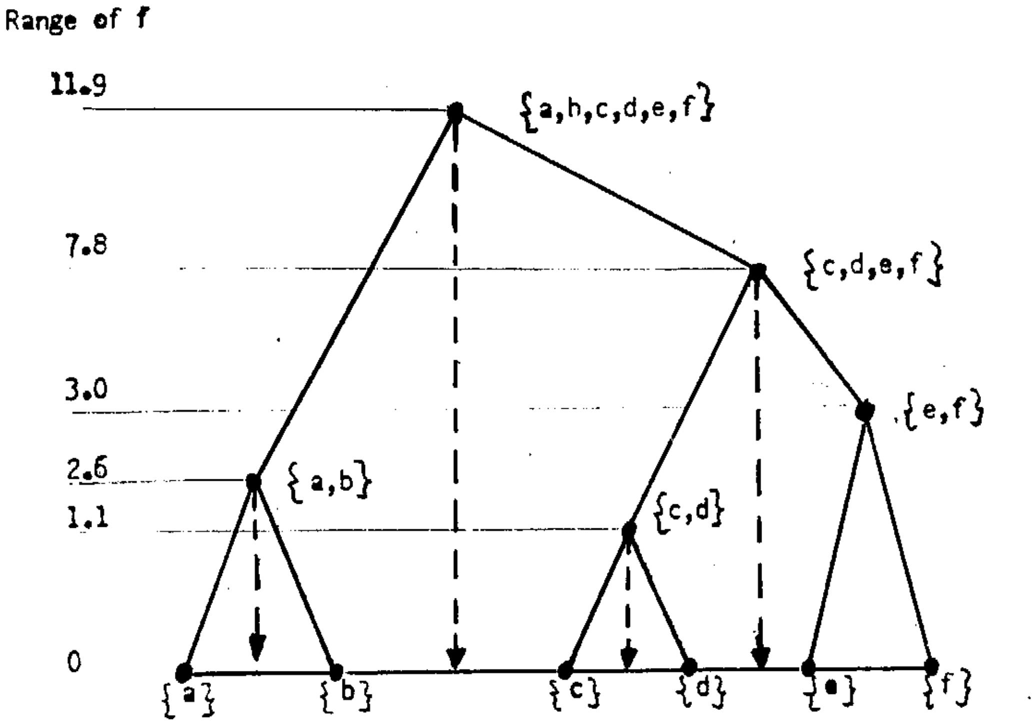


Figure 1. Tree and ultrametric path representation of a hierarchy on {a,b,c,d,e,f} .

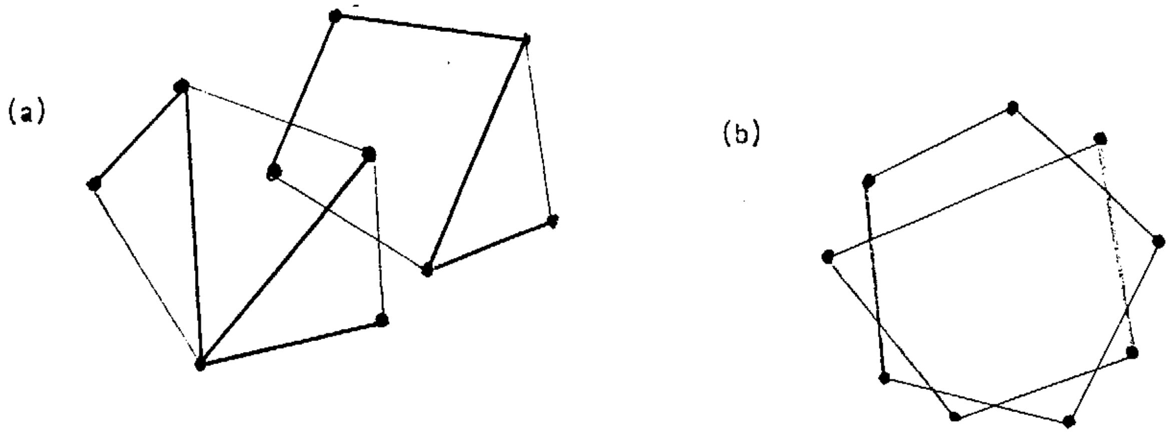


Figure 2. Illustratien of tree-connected admissibility (a) and not tree-connected admissibility (b).

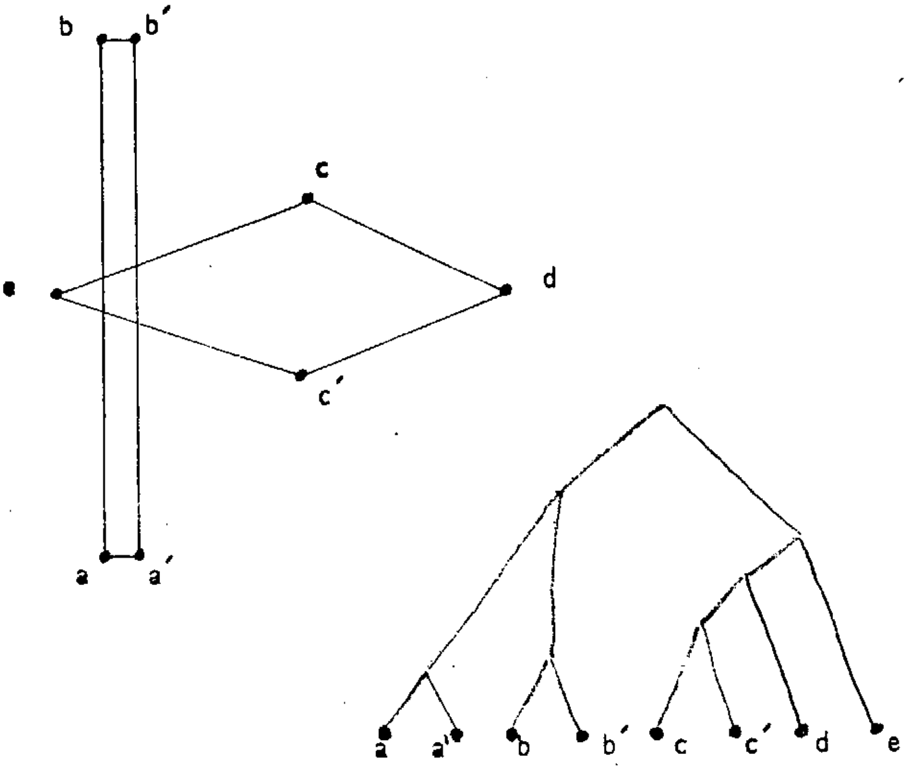


Figure 3. Complete linkage hierarchy can be not tree-connected admissible.

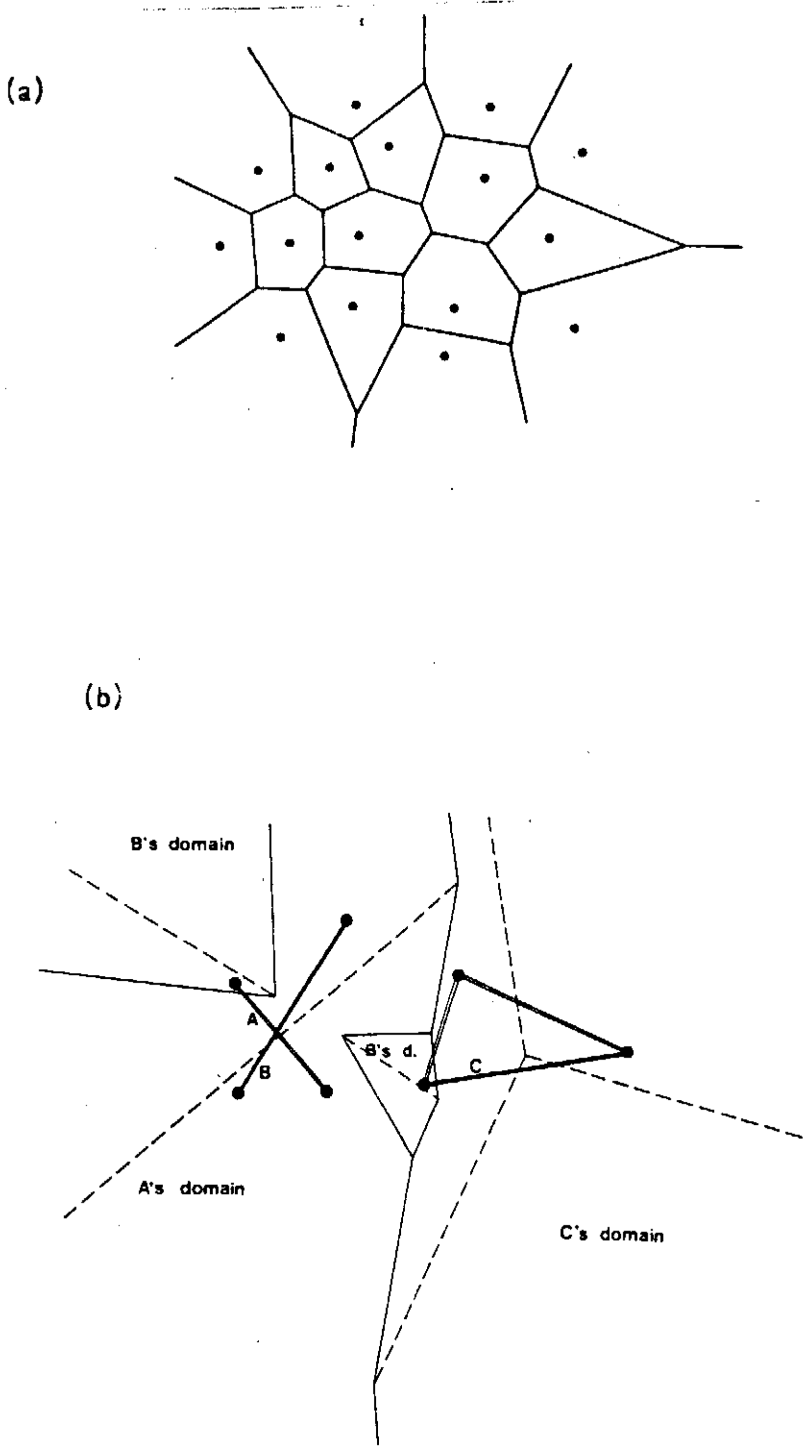


Figure 4. Voronoi diagram (a) and generalized Voronoi diagram for the complete linkage (b)