Fast Parallel Algorithm for the Single Link Heuristics of Hierarchical Clustering

Elias Dahlhaus
Basser Department of Computer Science
University of Sydney
NSW 2006, Australia

Abstract

A fast parallel algorithm of single link heuristics [10] of hierarchical clustering is presented. Its time processor product is optimal and the parallel time is the square of the logarithm.

1 Introduction

Hierarchical clustering has its applications in many fields of applied science. Examples are hierarchical classification schemes in biology, psychology, and pattern recognition problems. As long as the given distance function is an ultrametric the corresponding hierarchical classification is uniquely determined. For this case an optimal parallel algorithm has been developed [9]. Several sequential hierarchical clustering algorithms for the non ultrametric distance functions are known. An intensive study can be found in the book of Dubes and Jain [10]. Here we present a parallel algorithm for a special heuristics, the so called single link method. It is based on computing a minimum spanning tree which can be done in $O(\log^2 n)$ time using $O(n^2/\log^n)$ processors [8]. The main gap to be filled is to compute a hierarchical clustering tree (dendrogram) from a minimum spanning tree. We shall prove that this can be done in $O(\log n)$ time using $O(n)$ processors.

Therefore the overall time-processor product of $O(n^2)$ is optimal.

Section 2 introduces the fundamental notations and concepts. The single link heuristics will be presented. Section 3 presents the announced parallel algorithm to execute the single link heuristics.

2 Notations and Fundamental Definitions

2.1 The Problem of Hierarchical Clustering.

The problem of *hierarchical clustering* is the following:

Let d be *distance function* on a set V of items. That means d is a symmetric function from $V \times V$ into the set of positive real numbers such that $d(x,y) = 0$ iff $x = y$. The problem is to find a reasonable collection $\mathcal{C}$ of subsets of V, such that $\mathcal{C}$ is a *hierarchical classification* (*hierarchical clustering*). That means all pairs C_1, C_2 of sets in $\mathcal{C}$ are comparable by inclusion or disjoint. Moreover, we assume that the whole set V of items belongs to $\mathcal{C}$. The sets $C \in \mathcal{C}$ are called *clusters*. An item x is *more related to y than to z* iff there is a cluster $C \in \mathcal{C}$ such that x and y are in C but z is not in C.

The problem to be considered is to find a hierarchical classification such that the relation $d(x,y) < d(x,z)$ coincides with the relation "x is more related to x that to y" as much as possible.

One example of a hierarchical classification is the classification of species of animals.

We say that y is more related to x than z if there is a cluster $C \in \mathcal{C}$ such that x and y are in C but z is not in C. The relatedness should be as compatible with the distance function as possible.

2.2 Notions from trees

A *tree* is a connected graph which has no undirected cycles. A directed graph T is called a *root directed tree* if if its underlying undirected graph is a tree and the edges are directed in such a way that there is a vertex r (*root*) such that, for each vertex v of T, there is a directed path in T from v to r.

For any non root vertex v of a root directed tree T, the unique vertex w such that (v,w) is an edge in T is called the *parent* of v. The function P_T which matches to each non root vertex of T its parent is called the *parent function* of T. If v is the parent of a vertex w then w is called a *child* of v. If there is a directed path from v to w in the root directed tree T then w is called an *ancestor* of v and v is called a *descendent* of w. The least common ancestor of two vertices x and y of the root directed tree, denoted by $LCA(x,y)$, is the unique vertex v of T which is an ancestor of x and y and has the property that no proper descendent of v is an ancestor of both vertices, x and y. A vertex which has no children is called a *leaf*.

Let $\mathcal{C}$ be a hierarchical classification. Then we get the following tree T, called the *cluster tree* or *predendrogram* of $\mathcal{C}$:

The vertex set V_T consists of $V \cup \mathcal{C}$. (C_1, C_2) is an edge in T iff $C_1 \subset C_2$ and there is no C between C_1 and C_2. For a vertex $v \in C \in \mathcal{C}$ we set (v, C) to be an edge in T iff there is no $C' \in \mathcal{C}$, such that $x \in C' \subset C$.

It is easily seen that T defines a tree. Vice versa each root directed tree with V as leaf set and at least

0-8186-3200-3/92 $03.00 © 1992 IEEE

two children, for each non leaf vertex, defines hierarchical classification:

For each $t \in V_T$ let C_t be the set of leaves of the subtree T_t of T which consists of all descendents of t. Then $\mathcal{C} = \{C_t : t \in V_T \setminus V\}$ defines a hierarchical classification.

2.3 Notions from Metrics

We assume that V is a finite set of the form $\{1, \ldots n\}$. The distance function d is implemented as an $n \times n$ matrix.

Our aim is to compute a hierarchical classification $\mathcal{C}$ from a distance function d on V, such that, for $x, y \in C \in \mathcal{C}$, and $z \in V \setminus C$ the relation $d(x,y) < d(x,z)$ is satisfied reasonable often.

A distance function d is called an *ultrametric* iff the following *extended triangle inequality* is valid:

$$d(i,j) \leq \max\{d(i,k), d(j,k)\}.$$

Note that $+$ is replaced by max and that an ultrametric is ever a metric.

If d is an ultrametric then we ever get a hierarchical classification $\mathcal{C}$ such that x is more related to y than to z in $\mathcal{C}$ iff $d(x,y) < d(x,z)$, for all $x, y, z \in V$. This is a characteristic property of ultrametrics. Therefore each ultrametric can be expressed by a rooted directed vertex labeled tree (*dendrogram*) with the properties as defined below.

A *dendrogram* is a root directed tree T together with a positively real valued labeling h of the vertices, such that $h(v) < h(w)$ if w is an ancestor of v. We call $h(v)$ also the *height* or the *age* of v.

Proposition: (see for example [12]) A distance function d on V is an ultrametric iff there is a dendrogram (T, h) such that

1. V is the set of leaves of T and
2. for all $u, v \in V$, the distance $d(u,v)$ is the labeling $h(LCA(u,v)$ of the least common ancestor $LCA(u,v)$ of u and v with respect to T.

□

2.4 Notions from Complexity

The computation model is the concurrent read exclusive write parallel random access machine (CREW-PRAM) [11]. We assume that the comparison of two real numbers can be done in one time unit. Whenever it is written that a problem can be solved in time t with p processors it means that this problem can be solved in these bounds by a CREW-PRAM.

2.5 The Single Link Heuristics

The *single link heuristics* is defined by the following scheme (see [10]).

For each real number $k \leq \max_{i,j \in V} d(i,j)$, the *threshold graph* $G_k = (V, E_k)$ consists of the vertex set V and the edge set $E_k = \{ij : d(i,j) \leq k\}$. For each k, we make each connected component of G_k a cluster C. The ***single link classification*** or ***single link clustering*** corresponding to the distance function d is the set $\mathcal{C}_d = \{C | C$ is a connected component of some threshold graph G_k of $d\}$. It is well known that $\mathcal{C}_d$ is a hierarchical classification (see for example [10]).

3 The Parallel Single Link Algorithm

As in [13] we compute at first a minimum spanning tree (MST) T for the complete graph with vertices $1, \ldots, n$ and $d(x,y)$ as label for the edge xy. This can be done in $O(\log^2 n)$ time using $O(n^2/\log^2 n)$ processors [8].

Lemma 1: Let G_k be defined as above. Then x and y are in the same connected component of G_k iff all edges on the unique path from x to y in the MST T have a label of at most k.

Sketch of Proof: This follows from the easy observation that each MST T restricted to some connected component of G_k remains a tree, not only a forest.

□

We can express this lemma also as follows:

For each x and y, let $d'_T(x,y)$ be the maximum label of an edge on the unique path from x to y in T. Then x and y are in the same connected component of G_k iff $d'_T(x,y) \leq k$. We call d'_T the *maximum distance* metric of T. One can easily check that d'_T is an ultrametric and d'_T is independent on the choice of the MST T. d'_T is only dependent from d. We call d'_T also the ***single link ultrametric*** of d.

Moreover one can prove:

Lemma 2: (see for example [10]) The single link clustering $\mathcal{C}_d$ of d coincides with the hierarchical clustering corresponding to the ultrametric d'_T.

□

Our aim is to compute the dendrogram for d'_T, called also the *single link dendrogram* of d, by an efficient parallel algorithm.

We assume that the MST T is known.

To design a parallel algorithm for the single link heuristics, we construct a root directed tree T' with a distance function d' on the edges, called the *modified pseudo minimum spanning tree* of T and d. (MPMST). The essential properties of the MPMST are the following:

1. The maximum distance metric of T with respect to d and the maximum distance function of T' with respect to d' are equal.
2. Let $F(x)$ be the parent of x in T'. Suppose $F(x)$ is not the root of T'. Then $d'(x, F(x)) < d'(F(x), F(F(x)))$.

Afterwards we construct the dendrogram for the maximum distance metric of T' and d' which coincides with the dendrogram of the maximum distance metric of T and d and therefore with the dendrogram corresponding to the single link clustering of (V, d).

We continue with the construction of an MPMST.

We begin with the following observation:

Lemma 3: Let xy and zx be edges of the minimum spanning tree T and $d(x,y) \leq d(z,x)$. Consider the tree T' which arises from T by replacing the edge zx by the edge zy and labeling the edge zy by the distance $d(z,x)$. Then the maximum distance of T and the maximum distance of T' are equal.

□

From T and d, we can construct an MPMST T' as follows:

1. We direct T to a root r (by Eulerian circuit techniques [16])

2. For each x with y as its parent in T, let $F(x)$ be the ancestor a of y, such that all edges on the unique path from y to a have a distance of at most $d(x,y)$ and a is such an ancestor which is near to the root as possible. That means, for the parent b of a, the distance $d(a,b)$ is greater than $d(x,y)$ or $a = r$ and for all edges on the unique path from x to a have a distance of at most $d(x,y)$.

3. T' is the root directed tree with parent function F. The distance $d'(x,F(x)$ is $d(x,y)$ if y is the parent of x in T.

By iterated application of lemma 2, the maximum distances of the tree T' and the labeling d' are the same as the maximum distances in T with respect to the labeling d.

By construction, if $F(x)$ is a non root vertex (with respect to T and T') then $d'(x,F(x)) < d'(F(x),F(F(x)))$.

The root direction of T can be done in $O(\log n)$ time using $O(n)$ processors by [16].

The non trivial part is to fill out step 2 by a parallel procedure.

Let T be an MST and r be the root of T. For each vertex $v \neq r$, let $p(v)$ be the parent of v with respect to T and its root direction to r.

The computation of the parent function F of T' consists of two steps:

1. For each $k = 1,\ldots,\log n$, we compute a pointer F_k with $F_k(x) = p^{2^k}(x)$ and the maximum distances of x and $F_k(x)$.

2. With the help of the pointers F_k, we compute $F(x)$ by a binary search procedure.

The first step can be done in $O(\log n)$ time using $O(n)$ processors by standard methods: Note that $F_{k+1} = p^{2^{k+1}} = p^{2^k} \circ p^{2^k} = F_k \circ F_k$. Therefore p^{2^k} can be computed in time $k \leq \log n$ using n processors. Note that $p(x)$ is not defined if x is the root and $F_k(x)$ is not defined for those x with distance $< 2^k$ from the root. To compute the maximum distances $d'(x,F_k(x))$, we use the fact that $d'_T(x,F_{k+1}(x)) = \max(d'_T(x,F_k(x)), d'(F_k(x),F_k(F_k(x))))$. Therefore the distances $d'_T(x,F_{k+1}(x))$ can be computed from the distances $d'_T(x,F_k(x))$ in one time unit using $O(n)$ processors and, since k is bounded by $\log n$, all distances $d'_T(x,F_k(x))$ can be computed in $O(\log n)$ time using n processors.

Now we compute in parallel the tree T' by computing its parent function F. This is done by binary search using the distances $d'_T(x,F_k(x))$:

Algorithm Binary Search

Input: A root directed tree T, the functions $F_k(x)$ as defined above, the maximum distances $d'_T(x,F_k(x))$ for each $x \in V$ and each $k = 1,\ldots,\log n$.

Output: An MPMST T' for T and d together with a positive real valued labeling d' on the edges of T'.

1. For all $x \in V$, let k_x be the maximum k such that $d'_T(x,F_k(x)) \leq d(x,p(x))$ and $F_k(x)$ is defined and let $F'(x) = F_{k_x}(x)$

2. For $k = \log n$ down to 1:
   for all $x \in V$ in parallel:
   If $k_x \geq k$ and if $d'_T(F'(x),F_{k-1}(F'(x))) \leq d(x,p(x))$ then let
   $F'(x) = F_{k-1}(F'(x))$.

3. Let $F := F'$. T' consists of the directed edges $(x,F(x))$. Each edge $(x,F(x))$ is labeled by the distance $d'(x,F(x)) = d(x,p(x))$.

The correctness of algorithm **Binary Search** is proved as follows. After execution of 1, $F'(x)$ is of the form $p^{2^{k_x}}(x)$. Repeat that $F(x)$ is the $p^l(x)$ such that $d(p^l(x),p^{l+1}(x)) > d(x,p(x))$ such that l is minimal. This l is smaller than 2^{k_x+1} and at least 2^{k_x}.

It is easily checked, by backward induction on k, that in 2, after the execution of step k, $F(x)$ is an ancestor of $F'(x)$ or equals $F'(x)$ and $F(x)$ is a proper descendent of $F_{k-1}(F'(x) = p^{2^{k-1}}(F'(x))$. As long as $F'(x)$ and $F(x)$ do not coincide, one finds a $k' < k$ such that $d'_T(F'(x),F_{k'-1}(F'(x)) \leq d(x,p(x))$.

Therefore, after we leave step 2, $F(x)$ and $F'(x)$ coincide.

It remains to make a complexity analysis. The only steps which need more than constant time are the steps 1 and 2. These steps are loops of depth $\log n$ which need again constant time. It is easily seen that all steps need only a processor number of $O(n)$. Therefore we get the following result:

Theorem 4: Given an MST and the distances of the edges of the MST, we can construct its corresponding MPMST in $O(\log n)$ time using $O(n)$ processors.

The single link dendrogram for d (which coincides with the dendrogram for d'_T) can be computed from the MPMST T' by the following algorithm, which coincides with the algorithm in [9] which computes the dendrogram of an ultrametric if its "special minimum spanning tree" (the MPMST of an ultrametric) is known:

Algorithm Dendrogram

Input: An MPMST T' with its distance function d'.

Output: A dendrogram (T,h) consisting of a tree T and a height function h.

1. for each $x \in V$, let $C(x)$ be the set of children of x with respect to T', say
$$C(x) = \{y : F(y) = x\}.$$

2. for each $x \in V$, sort $C(x)$ by $d(y,x)$;
   for $y_1,y_2 \in C(x)$, let $y_1 \equiv y_2$ iff $d(y_1,x) = d(y_2,x)$,(or set $y_1 \equiv y_2$ iff $F(y_1) = F(y_2)$ and $d(y_1,F(y_1)) = d(y_2,F(y_2))$);
   for each $x \in V$, let $[x]$ be the $\equiv$-equivalence class x belongs to;

3. the vertex set of T consists of all $\equiv$-equivalence classes and all elements of V:

$$V_T = V \cup \{[x] : x \in V \text{ and } F(x) \text{ is defined}\}$$

4. for each $u \in V$, the height $h(u)$ of u is 0;
for each $\equiv$-equivalence class $[x]$, the height $h([x])$ is set $d(x, F(x))$;

5. directed edge set of T is defined by the following parent function Suc:

   - if $u \in V$, then $Suc(u)$ is the $[x]$, such that $x = u$ or $F(x) = u$ and $h([x]) = d(x, F(x))$ is minimal (We consider all T'-edges, such that u is incident. From these T'-edges we select one of minimal distance and let $Suc(u)$ be the equivalence class where the source of the selected edge belongs to);
   - if $u = [x]$ and $x \in C(y)$, such that $d(x, y)$ is maximal, then $Suc([x]) = [y]$;
     otherwise if $d(x, y)$ is not maximal, let $Suc([x])$ be the $[z]$, such that $z \in C(y)$ $(F(z) = F(x))$, $d(z, y) > d(x, y)$, and $d(z, y)$ is minimal under these conditions;

The algorithm **Dendrogram** needs $O(n)$ processors and $O(\log n)$ time, because sorting has this time bound [7]. In [9] it has been proved that the algorithm works correctly if F satisfies the the properties of an MPMST.

Concatenating all algorithms in the right way, we get the main result of this paper.

Main Theorem:

1. Given a minimum spanning tree of a distance function, its single link dendrogram (and therefore a cluster tree for its single link clustering) can be computed by a CREW-PRAM in $O(\log n)$ time using $O(n)$ processors.

2. Given a distance function, its single link dendrogram (and therefore a cluster tree for its single link clustering) can be computed in $O(\log^2 n)$ time using $O(n^2/\log n)$ processors.

4 Conclusion:

The single link clustering is not the only way of hierarchical clustering if the underlying distance function is not an ultrametric. It remains an open question to parallelize global link clustering (see [10]) or any other hierarchical clustering algorithm.

References

[1] B. Awerbuch, Y. Shiloach, *New connectivity and MSF algorithms for ultracomputer and PRAM*, Proceedings of the International Conference on Parallel Processing 1983, pp. 175-179.

[2] H. Bandelt, *Recognition tree metrics*, SIAM Journal on Discrete Mathematics 3 (1990), pp. 1-6.

[3] H. Bandelt, A. Dress, *Reconstructing the shape of a tree from observed similarity data*, Advances Applied Mathematics 7 (1986), pp. 309-343.

[4] F. Boesch, *Properties of the distance matrix of a tree*, Quaterly Applied mathematics 26 (1968), pp. 607-609.

[5] P. Buneman, *The recovery of trees from measures of dissimilarity*, in Mathematics in the Archaeological and Historical Sciences, F.R. Hodson et al. ed., Edinburgh University Press, Edinburgh, Scotland, 1971, pp. 387-395.

[6] P. Buneman, *A note on the metric properties of trees*, Journal of Combinatorial Theory Series B, vol. 17 (1974), pp. 48-50.

[7] R. Cole, *Parallel Merge Sort*, 27. IEEE-FOCS (1986), pp. 511-516.

[8] F. Chin, J. Lam , I. Chen, *Efficient Parallel Algorithms for some Graph Problems*, Communications of the ACM 25 (1982), pp. 659-665.

[9] E. Dahlhaus, *Fast Parallel Recognition of Ultrametrics and Tree Metrics*, to appear in SIAM Journal on Discrete Mathematics.

[10] A. Jain, R. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, New Jersey (1988).

[11] S. Fortune, J. Wyllie, *Parallelism in Random Access Machines*, 10. ACM-STOC (1978), S. 114-118.

[12] A. Gordon, *A Review of Hierarchical Classification*, Journal of the Royal Statistical Society A, vol. 150 (1987), pp. 119-137.

[13] J. Gower, G. Ross, *Minimum Spanning Trees and Single Linkage Cluster Analysis*, Applied Statistics 18 (1969), pp. 54-64.

[14] C. Kruskal, L. Rudolph, Marc Snir, *Efficient parallel algorithms for graph problems*, Algorithmica 5 (1990), pp. 43-46.

[15] B. Schieber, U. Vishkin, *On Finding Lowest Common Ancestors: Simplification and Parallelization*, SIAM Journal on Computing 17 (1988), pp. 1253-1262.

[16] R. Tarjan, U. Vishkin, *An Efficient Parallel Biconnectivity Algorithm*, SIAM-Journal on Computing 14 (1985), pp. 862-873.