On the Complexity of Ordinal Clustering

Rahul Shah[*] and Martin Farach-Colton[**]

Rutgers University, New Brunswick, NJ

Abstract. Given a set of pairwise distances on a set of n points, constructing an edge-weighted tree whose leaves are these n points such that the tree distances would mimic the original distances under some criteria is a fundamental problem. For example, this problem is sometimes called the *heirarchical clustering problem.*

One distance preservation criterion is to preserve the total order of pairwise distances. We show that the problem of finding a weighted tree, if it exists, which would preserve the total order on pairwise distances is NP-hard. A partial order on pairwise distances between points in which orders all distances that share an end point, so that each point has a view of the other points that is consistent with the original distances, is called a *triangle order*, since it is equivalent to an order where this distances in each triangle are ordered. This order has been studied in biological settings. We also show the NP-hardness of the problem of finding the weighted tree which would preserve a triangle order.

1 Introduction

Clustering data based on pairwise distances is a fundamental problem. Weighted trees can be used to represent heirarchical clusters. Thus, constructing weighted trees that "fit" a distance matrix is a type of clustering, and this metric based problem has been extensively studied, for example in evolutionary biology and computational linguistics The general problem is to find an edge weighted tree which approximates the distance matrix under some criteria. These kind of trees are also known as *phylogenetic trees* or simply *phylogenies.* A particular instance of this problem has been considered in the algorithmic computational biology community : Given a (total or partial) order on the pairwise distances between points, give an edge weighted tree on those points so that the pairwise pathlengths between these points in the tree satisfies this order. This can also be seen as the ordinal version of the Heirachical Multidimensional Scaling (HMDS) problem [5]. If the tree is unweighted then pairwise pathlengths would be number of edges on the path. The problem of obtaining such an unweighted tree was considered by Kannan and Warnow [1] and Kearney, Hayward and Meijer [2].

In general, we are given an $n \times n$ distance matrix M and asked to find tree T on leaves $1, 2, .., n$ such that path distance $d_T(i, j)$ in the tree closely approximates the matrix M. When $d_T(i, j) = M(i, j)$ the matrix is said to

[*] Dept. of CS, Rutgers University, sharahul@paul.rutgers.edu

[**] Google Inc., martin@google.com

be *additive* and efficient algorithms exist for constructing tree from additive distance data in $O(n^2)$ time [9, 10, 12]. But when the matrix is not additive then various optimization criteria were proposed and many of them are shown to be NP-hard [7, 2, 8].

However, for many applications, the actual numeric data is quite unreliable (see [8]). The distance data obtained through experiments could have kinds of errors like *homoplasy* and *superimposed changes* [2]. Also there are experiments which give only relative information about pairwise distances [1]. Hence, Kannan and Warnow [1] took an approach which assumes confidence in only relative information. So the input is in the form of partial or total order on the pairwise distances. [2] and [1] consider the problem of constructing an edge-weighted tree which preserves the order among the distances. The advantage of using orders is that they are less vulnerable to errors due to superimposed changes. Superimposed changes do not affect the relative ordering of pairwise distances.

The first explicit study of ordinal methods for inferring phylogenetic trees was undertaken by Kannan and Warnow [1]. They gave an $O(n^3)$ algorithm for constructing an unweighted tree , if it exists, from a triangle order on pairwise distances. In the unweighted case, each edge of the tree is assumed to have unit weight. And also it is assumed that there are no vertices with degree two. Kearney, Hayward and Meijer [2] extended this work. They gave an $O(n^2 \log^2 n)$ algorithm for constructing an unweighted tree, if it exists, from a total order. The weighted case was posed as open problem in both the papers. The unweighted case is a special case of weighted case. Although we have polynomial time algorithms to find the unweighted tree, for many orders the unweighted tree representing that order may not exist at all even if the weighted tree exists. A polynomial time algorithm was conjectured in these papers for the weighted case. The algorithm was based on so called *mid-path tree conjecture* which stated that if the weighted tree representing triangle or a total order existed then it could be represented with some weight function on a special tree called *midpath tree* corresponding to that order. Contrary to the conjecture, we show in this paper that the problem of constructing weighted trees from total as well as triangle orders is NP-hard. We call these problems *Total Ordinal Clustering* and *Triangle Ordinal Clustering*, respectively. The reduction uses the generalization of a counter-example to the midpath tree conjecture given by Shah and Farach-Colton [5]. This NP-hardness is an interesting result since the unweighted case here is polynomial time solvable. Also, these are the first NP-hard results in this area where all the known NP-hard problems come only from incomplete distance matrices or incompletely specified orders.

In section 2 we give some concepts and basic lemmas established in the field. In section 3, we show a generalization of the counter example for Midpath Tree Conjecture In section 4, we show the NP-hardness of the Triangle Ordinal Clustering problem. In section 5, we extend the NP-hardness result of section 4 to Total Ordinal Clustring. In section 6, we comment on ordinal clustering in metric spaces other than trees and we conclude with future work in section 7.

2 Preliminaries

Here, we discuss some preliminary concepts and results known in this area. We shall also introduce some useful concepts that we will use in proving our main results.

2.1 Realizability and LP's

Define d_{T_w} to be the distance metric of a tree T under a non-negative weight function w, where $d_{T_w}(s,t)$ is sum of the weights of the edges on the unique path $P_T(s,t)$ from leaf s to leaf t in T. A partial order P on pairwise distances is said to be *realizable* as tree T if, for some weight function w on the edges of T, we get $d_{T_w}(a,b) \leq d_{T_w}(c,d)$ whenever $d(a,b) \leq_P d(c,d)$

Given a tree T, we can determine in polynomial time via linear programming whether or not the partial (or total) order P can be realized as T by checking the feasibility of the order constraints

$$d_{T_x}(c,d) - d_{T_x}(a,b) \geq \delta \text{ if } d(a,b) <_P d(c,d) \tag{1}$$

$$d_{T_x}(c,d) - d_{T_x}(a,b) = 0 \text{ if } d(a,b) =_P d(c,d) \tag{2}$$

$$x \geq 0 \tag{3}$$

where $\delta > 0$ is a constant.

2.2 Contractions and Expansions

A *contraction* of a tree T at the edge pq is the tree that results from removing edge pq from T and identifying vertices p and q. An *expansion* of T at a vertex v is the inverse operation of contraction,though many expansions may be possible at any node. A tree $T^{'}$ is called a contraction of T if $T^{'}$ is obtained by the contraction of T at some edge, or if $T^{'}$ is a contraction of some contraction of T. $T^{'}$ is called an expansion of T if T is a contraction of $T^{'}$.

2.3 Midpath Trees, Triangle order and MPT conjecture

Many assertions in this subsection are analogically true for triangle as well as total orders. In fact they would be true for any partial order at least as specified as triangle order. We call such orders as *supertriangular order.* Note that triangle order and total order are two extreme subcases of supertriabgular order. Let T be a tree which realizes a supertriangular order Δ. For any two leaves x and y, the midpoint of the weighted path $P_T(x,y)$ is that point on $P_T(x,y)$ which is equidistant from x and y. Let u denote the edge or the vertex on which this midpoint is located. Consider $T' = T - u$. T' consists of at least two connected components. Now, since T realizes the order following trichotomy property holds:

- For all leaves v in the connected component of T' containing x, $d(v,x) <_\Delta d(v,y)$.

- For all leaves v in the connected component of T' containing y, $d(v,y) <_{\Delta} d(v,x)$.
- For all leaves v not in the connected components of T' containing x or y, $d(v,x) =_{\Delta} d(v,y)$.

The edge or vertex on $P_T(x,y)$ that satisfies the trichotomy property is called the *midpath* of $P_T(x,y)$ and is denoted by $M(x,y)$.

Given a tree T which realizes a triangle (or total) order Δ on pairwise distances between points in set S, along with a weight function w,consider a function $m : S \times S \rightarrow E(T) \bigcup V(T)$ which maps each pair of leaves a, b to the midpath $M(a,b)$ on which the midpoint of the weighted path $P_T(a,b)$ falls. Now, contract all the (non-leaf)edges in T which do not have any midpaths falling on them to obtain an unweighted tree. This is the tree on which midpaths for all pairs of leaves exists and these midpaths satisfy the trichotomy property. We call this tree a *midpath tree* T_{Δ}. Note that a midpath tree is just an unweighted tree with the midpath function.

The midpath $M(a,b)$ gives a bipartition (or tripartition) of points in S based on whether they are closer to a or to b (or equidistant). Given the midpath tree with the Midpath function, we can construct the triangle order represented by it. This means the midpath tree T_{Δ} and midpath function M (weight function not required) can be used to represent a unique triangle (*not* total) order. The midpath tree is a minimal tree on which midpath function satisfying such a trichotomy property can be defined. As noted in the beginning of this subsection, if a supertriangular order can be realized by some tree T, then the midpath tree for this order exists. That means the existence of the midpath tree is a prerequisite for the existence of a tree T which realizes the supertriangular order order. Hence, while we don't know the algorithm to construct a tree T realizing the supertraingular order, the algorithm to construct the midpath tree is considered to be the first step in developing any such algorithm. and any such T is an expansion of T_{Δ} [2]. Intuitively, the existence of midpath tree indicates that the supertriangular order has a tree-like nature.

Following two lemmas (for proofs see [2]) show the importance of midpath trees.

Lemma 1. *If it exists, the midpath tree of a supertriangular order is unique.*

Lemma 2. *If T is a tree that realizes a supertriangular order Δ, then T is an expansion of midpath tree T_{Δ}.*

The following is an $O(n^3)$ algorithm that constructs the midpath tree: begin with a star topology and repeatedly expand until all midpaths exists. An optimal $O(n^2)$ algorithm for finding the midpath tree was given by [6]. Hence, constructing the midpath tree is considered to be the first step in finding the tree T that realizes the supertriangular order. Kearney, Hayward and Meijer [2] start with the midpath tree and then give an algorithm to expand it to obtain an unweighted tree that represents the total order, if such a tree exists.

To summarize, Given a weighted tree, the distance matrix can be can be constrcuted from it using pair wise pathlengths. Given a distance matrix, a total order can be induced by it and then the triangle order can be induced as a subset of this total order. Given a weighted tree, the midpath tree can be constructed from it and again triangle order can be constructed from the midoath tree. If a midpath tree exists, it can be constrcuted from the triangle order and hence from total order and hence form distance matrix. Lemma 3 gives and example of the midpath tree induced by a total order.

To construct the weighted tree representing the supertriangular order, it was not known whether we need to expand the midpath tree or if the midpath tree itself is the tree T that realizes the order. In case an expansion is necesary, no general methods for expanding the midpath tree are not known. This led to the conjecture [3]:

"If a supertriangular order Δ is realizable as some tree, then it is realizable as the midpath tree T_Δ"

If this conjecture were to be true, then no expansion methods would be necessary. We could just construct a midpath tree and run the linear program given by (1), (2), (3) and it would give us the required weighted tree. If the linear program were infeasible, then we would know that no tree realizes the order. But this conjecture is false for both total and triangle orders. In the following subsections, we shall show the counter-examples.

2.4 LP on Midpath tree and Dual

If the midpath tree conjecture were true, it would mean that if a triangle order $\leq_\Delta$ is realizable, it can be realized by assigning weights to the edges of mid-path tree which satisfy the following linear constraints (5),(6) and (7).

Let $P_T^+(a,b)$ = set of edges on the path from a to mid-point $M(a,b)$, including the edge $M(a,b)$ if $M(a,b)$ is an edge. Let $P_T^-(a,b)$ be the set of edges on the path from $M(a,b)$ to b not including $M(a,b)$. For each e dge e, let x_e be weight variable on edge e.Then $\forall a, b \in S$,

$$minimize\ 0 \tag{4}$$

$$\sum_{e \in P_T^+(a,b)} x_e - \sum_{e \in P_T^-(a,b)} x_e \geq \delta\ if\ M(a,b) \in E(T) \tag{5}$$

$$\sum_{e \in P_T^+(a,b)} x_e - \sum_{e \in P_T^-(a,b)} x_e = 0\ if\ M(a,b) \in V(T) \tag{6}$$

$$x_e \geq 0\ \forall e \tag{7}$$

Now let's see what the dual looks like. Let y_{ab} be the weight for the midpath constraint (a,b) in above LP. Then the dual is,

$$maximize\ \delta \sum_{M(a,b) \in E(T)} y_{ab} \tag{8}$$

$$\sum_{(a,b) ste \in P_T^{+}(a,b)} y_{ab} - \sum_{(a,b) ste \in P_T^{-}(a,b)} y_{ab} \leq 0 \ \forall e \tag{9}$$

$$y_{ab} \geq 0 \tag{10}$$

Then, to prove the infeasibility of this LP, we have to prove that its dual has unbounded maximum, because dual is always feasible with solution $y_{ab} = 0 \ \forall a, b$. It can be seen from the dual that if the maximum objective value is non-zero then it has unbounded maximum. So, any multiset of constraints from 5, 6 such that atleast one of them is from 5, which adds up to have nonpositive weight on each edge, forms the witness of infeasibility of the LP. To generate, the counter-example to midpath tree conjecture we need to work with the weighted tree, in which there is such a multiset of midpath constraints which add up positively only on some non-midpath edge. (Note that if it is a weighted tree then, any such multiset of constraints has to add up positively on atleast one edge.)

2.5 Lemmas and Counter-Examples

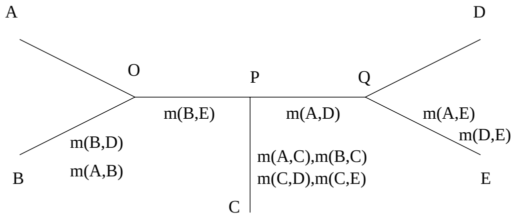


Fig. 1. Not all total orders are realizable

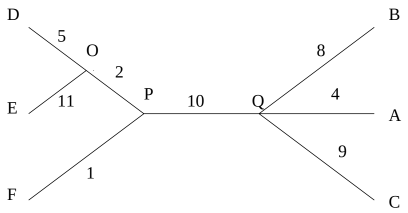


Fig. 2. Total orders not realizable as mid-path tree

Lemma 3. *There are total orders which are not realizable as any tree, despite the existence of midpath tree. [4]*

Proof : Consider the total order $AD < DE < AB < BD < AE < BE < AC < CD < CE < BC$. Figure 1 shows the mid-path tree for this total order. The mid-path tree is binary so it can not be expanded further. Following analysis shows that this tree can not be weighted to realize this total order.

$$BD < AE => BO + DQ < AO + EQ$$

$$AC < CD => AO + OP < PQ + DQ$$

$$CE < BC => PQ + EQ < BO + OP$$

summing up we get, $0 < 0$, which is contradiction. □

Lemma 4. *There are total orders which are realizable as some tree T but are not realizable as mid-path tree. [4]*

Proof : Figure 2 gives the example. The total order is the one generated by pairwise distances in the weighted tree shown in the figure. Note that no mid-point falls on the edge OP i.e. $\forall x, y\ M(x, y) \neq OP$. So mid-path tree is obtained by contraction at edge OP. Let's identify O and P both by R to obtain the mid-path tree. T hen,

$$AC < EF \Rightarrow AQ + CQ < ER + FR$$

$$DE < BC \Rightarrow DR + ER < BQ + CQ$$

$$CF < AD \Rightarrow CQ + FR < AQ + DQ$$

$$BE < CE \Rightarrow BQ < CQ$$

summing up , we get $0 < 0$. This is contradiction. Hence, this total order can not be realized as mid-path tree. □

Lemma 5. *There are triangle orders which are realizable as some tree but not the midpath tree. [5]*

Proof : Figure 3 gives the counter-example for the midpath tree conjecture for triangle order. Although much smaller counter example for total orders exists, note that the same forms the counter example for midpath tree conjecture for total orders.

It consists of a weighted tree T which is a realization of the obvious triangle order generated by the pairwise path distances in T. It consists of 18 leaves, and no midpoint falls on edge op i.e. $\forall x, y\ M(x, y) \neq op$. Hence edge op is contracted in the midpath tree. All other edges have some midpoints on them. The question is : *is it possible to have some other weight assignment of T which will maintain the same triangle order and will have weight of edge $op = 0$?* The linear program generated is infeasible, thus establishing that the answer is no. The table shows the witness to the infeasibility. Adding up the last column of the table we get,

$$pd + pa + oj + og < oc + oi + ob + oh$$

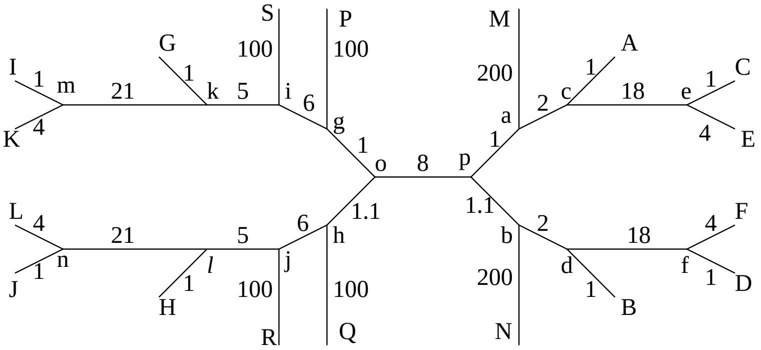


Fig. 3. Δ not Realizable as Midpath Tree

By a similar analysis, we can show that

$$pc + pb + oi + oh < od + oj + oa + og$$

Summing yields $op > 0$. Thus, the midpath tree cannot realize Δ even though some expansion of the midpath tree can.

pair	midpoint	inequality	that implies
AD	df	$AB <_\Delta BD$	$oA + 2pd < oD$
DE	ac	$DM <_\Delta ME$	$oD + 2pa < oE$
EG	ac	$EA <_\Delta AG$	$oE - 2oc < oG$
GJ	jl	$GR <_\Delta RJ$	$oG + 2oj < oJ$
JK	gi	$JP <_\Delta PK$	$oJ + 2og < oK$
KD	gi	$KS <_\Delta SD$	$oK - 2oi < oD$
DH	pb	$DN <_\Delta NH$	$oD - 2ob < oH$
HA	oh	$HQ <_\Delta QA$	$oH - 2oh < oA$

□

Lemma 6. *There are triangle orders which are not realizable at all, despite the existence of midpath tree.*

Proof : : Consider the midpath tree of triangle order in figure 3. This means we have to contract the edge op. Now, we change this triangle order slightly. We expand edge op vertically. And all the symmetric midpaths (e.g. M(GH), M(PQ), M(CD), ...) which were on the edges oh or pb are now moved to this new vertical edge op. Note that this midpath tree is binary, so it cannot be expanded any further. The witness of infeasibility in the proof of lemma 5, holds for infeasibility of this midpath tree too. This proves our lemma. □

2.6 Unrooted Quartet Consistency(UQC)

We present here a related problem of constructing trees using unrooted quartets, where a quartet is an unrooted tree on four leaves, v_i, v_j, v_k, v_l. Each quartet q is constrained to contain an edge e so that $q - e$ describes a partition descirbes a partition of the four leaves into two sets of two leaves each. We indicate this by writing $q = (v_i v_j, v_k v_l)$. The unrooted quarted consistency (UQC) problem is as follows.
Problem: Unrooted Quartet Consistency.
Input: A set Q of quartets on the set of points $S = \{v_1, v_2, ..., v_n\}$.
Question: Does there exist a tree T_Q with leaves labeled by points in S such that if $q = (v_i v_j, v_k v_l) \in Q$, then there is an edge e in T_Q such that v_i, v_j are on one side of e and v_k, v_l are on the other side.

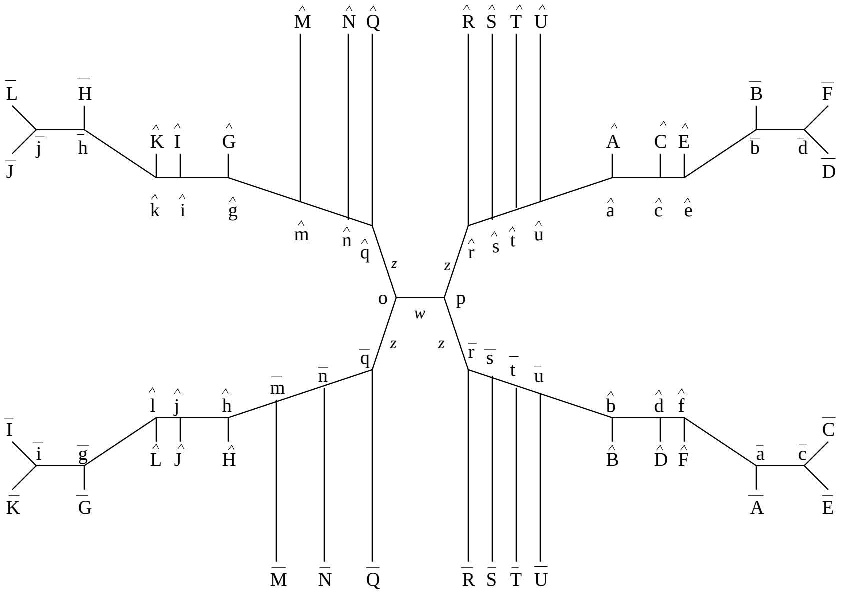
The UQC problem was shown to be NP-complete by Steel [11]. This is shown by reduction to betweenness problem. We shall use this problems to show the NP-hardness of triangle ordinal clustering as well as total ordinal clustering. The counter-example of lemma 5, imposes a quartet constraint and shows what edge expansion is needed. The main idea is to superimpose such counter-exmaples with different so as to generate the required quartet constraints. We need to make sure that these set of counter-examples do not interfere with each other i.e. remaining part of the order (the part not in witness) should stay in the same order after expansion. For generating such weight values for these set of trees we need well-separted numbers of the following section.

2.7 Well-Separated Numbers

We call set of m integers, $a_1, a_2, ..., a_m$, well-separated if each pair of these number produces a unique sum and $\forall i \; a_i \leq \; poly(m)$. These numbers could be obtained as $2m^2 i + i^2$. These numbers are composed of two parts $2m^2 i$ and i^2, if two pairs of numbers have same sum in one part, they can not have the same sum in the other part. And in sense of addition, the two parts of the numbers are scaled appropriately so that they do not interfere. Similarly, we define k-weighted set of well-separated integers $a_1, a_2, ...a_m$ if for all $u, v \leq k$, $ua_i + va_j$ is unique for each quadruple u, v, i, j and $\forall i a_i \leq \; poly(m)$. Note that k is considered to be a constant here. These numbers could be described as base $2k$ numbers. The numbers a_i consists of lower $\log m$ digits as the bits in binary representation of i, $(\log m + 1)$-th digit as 1 and leftmost part as $((\log m + 2)$-th digit onwards) $4m^2 ki + i^2$. That is, if $b_{\log m}, ..., b_1, b_0$ is the binary representation of i, then a_i is $\sum_{0 \leq j \leq \log m} (2k)^j b_j + (2k)^{\log m + 1} + (2k)^{logm+2}(4m^2 ki + i^2)$. Given $ua_i + va_j$ we can uniquely determine u, v from the right $\log m + 1$ digits and then a_i, a_j can be uniquely determined from the remaining part. Hence, each weighted sum is unique. These numbers are polynomial in m although exponential in the highest weight k.

We shall be using these numbers to construct the weights in polynomial time reduction to show the NP-hardness results. This requires the numbers not to be super-exponential. However, numbers exponential in m are would still maintain the polynomial time reducibility. We choose to stick with the polynomial weights though, due to its relation with unweighted (unit-weighted) case. This would imply that if the phylogenetic tree were allowed to have degree two vertices, then even the unit weighted case would be NP-hard.

3 Generalized Counter-Example to Midpath Tree Conjecture



$\hat{M}$ $\hat{N}$ $\hat{Q}$
$\hat{R}$ $\hat{S}$ $\hat{T}$ $\hat{U}$
$\bar{L}$ $\bar{H}$
$\bar{j}$ $\bar{h}$
$\bar{J}$
$\hat{K}$ $\hat{I}$ $\hat{G}$
$\hat{k}$ $\hat{i}$ $\hat{g}$
$\hat{m}$ $\hat{n}$ $\hat{q}$
$\hat{A}$ $\hat{C}$ $\hat{E}$
$\bar{B}$ $\bar{F}$
$\bar{b}$ $\bar{d}$
$\bar{D}$
$\hat{a}$ $\hat{c}$ $\hat{e}$
$\hat{r}$ $\hat{s}$ $\hat{t}$ $\hat{u}$
z z
o w p
z z
$\bar{q}$ $\bar{n}$ $\bar{m}$
$\bar{r}$ $\bar{s}$ $\bar{t}$ $\bar{u}$
$\hat{l}$ $\hat{j}$ $\hat{h}$
$\hat{b}$ $\hat{d}$ $\hat{f}$
$\bar{I}$ $\bar{i}$ $\bar{g}$
$\hat{L}$ $\hat{J}$ $\hat{H}$
$\hat{B}$ $\hat{D}$ $\hat{F}$
$\bar{C}$ $\bar{a}$ $\bar{c}$
$\bar{K}$ $\bar{G}$
$\bar{A}$ $\bar{E}$
$\bar{M}$ $\bar{N}$ $\bar{Q}$
$\bar{R}$ $\bar{S}$ $\bar{T}$ $\bar{U}$

Fig. 4. Δ not Realizable as Midpath Tree

Figure 4 gives a generalized couter-example for mid-path tree conjecture.It consists of a weighted tree T which is a realization of the obvious triangle order generated by the pairwise path distances in T. The weights on the tree are according to the following tables. Here, we show the distances of leaves from repspective points o or p for the convenience of analysis. This could fully determine all the edge weights if needed. The weights are in terms of parameters n, m, i which would be defined in the next section.

$$op = w$$

$p\hat{A} = x$	$p\hat{B} = x + 2z + 1$
$o\hat{G} = x + 4z + 2 + w$	$o\hat{H} = x + 6z + 3 + w$
$p\hat{C} = x + 8z + 4 + 2w$	$p\hat{E} = x + 14z + 8 + 2w$
$p\hat{D} = x + 10z + 5 + 2w$	$p\hat{F} = x + 12z + 7 + 2w$
$o\hat{I} = x + 16z + 9 + 3w$	$o\hat{K} = x + 20z + 12 + 3w$
$o\hat{J} = x + 18z + 10 + 3w$	$o\hat{L} = x + 22z + 13 + 3w$

Similarly, for distances of $\bar{A}$ to $\bar{L}$ replace x by y.

$$p\hat{r} = p\bar{r} = o\hat{q} = o\bar{q} = z$$

$p\hat{s} = p\bar{s} = 2z + 1.25$	$p\hat{t} = p\bar{t} = 3z + 2.25$	$p\hat{u} = p\bar{u} = 5z + 3.25$

$o\hat{n} = z + 0.75$	$o\hat{m} = 5z + 3.75$	$o\bar{n} = 3z + 1.75$	$o\bar{m} = 7z + 4.75$

for $\hat{A},, \hat{J}$ and $\bar{A}, \ldots, \bar{J}$ in (alphabetical) order $\hat{a}\hat{A} = 1$. Also,

$\hat{m}\hat{M} = \hat{n}\hat{N} = \hat{q}\hat{Q} = 10000nm + 400ni + 40n$
$\bar{m}\bar{M} = \bar{n}\bar{N} = \bar{q}\bar{Q} = 10000nm + 400ni + 80n$
$\hat{r}\hat{R} = \hat{s}\hat{S} = \hat{t}\hat{T} = \hat{u}\hat{U} = 10000nm + 400ni + 120n$
$\bar{r}\bar{R} = \bar{s}\bar{S} = \bar{t}\bar{T} = \bar{u}\bar{U} = 10000nm + 400ni + 160n$

For parameters, we choose $z = 4n, w = 1, x = 2000ni, y = x + 500n$. For the purpose of this section assume n is much bigger than 1. i could take any value between $1,, n$. Although, w is chosen to be 1, for any $1 \leq w \leq n$ this counter example is still valid and it maintains the same order. Parameters n and m will be useful in the next section and will be introduced then. The counter-example consists of 38 leaves, and no midpoint falls on edge op i.e. $\forall x, y \; M(x, y) \neq op$. Hence edge op is contracted in the midpath tree. Also, no midpoint fall on edges $o\bar{q}, o\hat{q}, p\bar{r}, p\hat{r}$. All other edges have some midpoints on them. The question is : *is it possible to have some other weight assignment of T which will maintain the same triangle order and will have weight of edge $op = 0$?* The linear program generated is infeasible, thus establishing that the answer is no. Following table shows the witness to the infeasibility.

pair	midpoint	inequality	that implies
$\hat{A}\hat{D}$	$\bar{u}\hat{b}$	$\hat{A}\bar{U} < \bar{U}\hat{D}$	$o\hat{A} + 2p\bar{u} < o\hat{D}$
$\hat{D}\hat{E}$	$\hat{s}\hat{t}$	$\hat{D}\hat{S} < \hat{S}\hat{E}$	$o\hat{D} + 2p\hat{s} < o\hat{E}$
$\hat{E}\hat{G}$	$\hat{t}\hat{u}$	$\hat{E}\hat{U} < \hat{U}\hat{G}$	$o\hat{E} - 2o\hat{u} < o\hat{G}$
$\hat{G}\hat{J}$	$\bar{m}\hat{h}$	$\hat{G}\bar{M} < \bar{M}\hat{J}$	$o\hat{G} + 2o\bar{m} < o\hat{J}$
$\hat{J}\hat{K}$	$\hat{n}\hat{m}$	$\hat{J}\hat{N} < \hat{N}\hat{K}$	$o\hat{J} + 2o\hat{n} < o\hat{K}$
$\hat{K}\hat{D}$	$\hat{n}\hat{m}$	$\hat{K}\hat{M} < \hat{M}\hat{D}$	$o\hat{K} - 2o\hat{m} < o\hat{D}$
$\hat{D}\hat{H}$	$\bar{r}\bar{s}$	$\hat{D}\bar{S} < \bar{S}\hat{H}$	$o\hat{D} - 2o\bar{s} < o\hat{H}$
$\hat{H}\hat{A}$	$\bar{q}\bar{n}$	$\hat{H}\bar{N} < \bar{N}\hat{A}$	$o\hat{H} - 2o\bar{n} < o\hat{A}$

pair	midpoint	inequality	that implies
$\hat{B}\hat{C}$	$\hat{t}\hat{u}$	$\hat{B}\hat{T} < \hat{T}\hat{C}$	$o\hat{B} + 2p\hat{t} < o\hat{C}$
$\hat{C}\hat{F}$	$\bar{s}\bar{t}$	$\hat{C}\bar{S} < \bar{S}\hat{F}$	$o\hat{C} + 2p\bar{s} < o\hat{F}$
$\hat{F}\hat{H}$	$\bar{s}\bar{t}$	$\hat{F}\bar{T} < \bar{T}\hat{H}$	$o\hat{F} - 2o\bar{t} < o\hat{H}$
$\hat{H}\hat{I}$	$\hat{m}\hat{g}$	$\hat{H}\hat{M} < \hat{M}\hat{I}$	$o\hat{H} + 2o\hat{m} < o\hat{I}$
$\hat{I}\hat{L}$	$\bar{n}\bar{m}$	$\hat{I}\bar{N} < \bar{N}\hat{L}$	$o\hat{I} + 2o\bar{n} < o\hat{L}$
$\hat{L}\hat{C}$	$\bar{n}\bar{m}$	$\hat{L}\bar{M} < \bar{M}\hat{C}$	$o\hat{L} - 2o\bar{m} < o\hat{C}$
$\hat{C}\hat{G}$	$\hat{r}\hat{s}$	$\hat{C}\hat{S} < \hat{S}\hat{G}$	$o\hat{C} - 2o\hat{s} < o\hat{G}$
$\hat{G}\hat{B}$	$\hat{q}\hat{n}$	$\hat{G}\hat{N} < \hat{N}\hat{B}$	$o\hat{G} - 2o\hat{n} < o\hat{B}$

table [] implies,

$$p\bar{u} + p\hat{s} + o\bar{m} + o\hat{n} < o\hat{u} + o\hat{m} + o\bar{s} + o\bar{n}$$

table [] implies,

$$p\hat{t} + p\bar{s} + o\hat{m} + o\bar{n} < o\bar{t} + o\bar{m} + o\hat{s} + o\hat{n}$$

adding up,

$$p\hat{t} + p\hat{u} + p\hat{s} + p\bar{s} < o\hat{u} + o\bar{t} + o\bar{s} + o\hat{s}$$

replacing $\hat{A}$ by $\bar{A}$, and similarly,

$$p\bar{t} + p\bar{u} + p\bar{s} + p\hat{s} < o\bar{u} + o\hat{t} + o\hat{s} + o\bar{s}$$

adding up,

$$op > 0$$

This implies that we need to expand the midpath tree, in order to realize the triangle order. Also, this means that any expansion which realizes the triangle order must have an non-zero weighted edge *op*, which imposes, to obtain the tree which realizes the triangle order, the expansion of the mid-path tree in which there is an edge which separates leaves $\hat{A}$, $\hat{B}$ from leaves $\hat{G}$, $\hat{H}$. In this sense, this (counter-) example imposes a quartet constraint on leaves $\hat{A}$, $\hat{B}$, $\hat{G}$, $\hat{H}$ such that if the triangle order was realizable then there must be a quartet separating edge e such that $T - e$, i.e. tree T cut at the edge e will have $\hat{A}$, $\hat{B}$ in one component and $\hat{G}$, $\hat{H}$ in the other. Also, the added advantage of using this (counter-) example over the previous is that this allows a region around the quartet edge (op) where no midpoint falls. This can be used by other super-imposed quartet constraints, to impose further expansions of the tree. We shall see this in the next section.

4 NP-hardness of Triangle Ordinal Clustering(TOC)

In this section we show that given a triangle order Δ it is NP-hard to determine whether there exists a weighted tree T which realizes Δ. Before we show the reduction, we introduce few definitions. Let T be the weighted tree in the genrelized counter example of the previous section.

Definition 1 *The subtree of T defined by cutting T at edge $p\hat{r}$ and taking the component which contains $\hat{A}$ is called $A-leg$. $B-leg, G-leg, H-leg$ are similarly defined. Note that these are weighted trees which follow the edge weights of T.*

Definition 2 *The leaves $\hat{M}, \hat{N}, \hat{Q}, \hat{R}, \hat{S}, \hat{T}, \hat{U}, \bar{M}, \bar{N}, \bar{Q}, \bar{R}, \bar{S}, \bar{T}, \bar{U}$ are classified as dummy points. While others are non-dummy points.*

Definition 3 *Given a weighted tree T on the vertex set V and a subset S of V (V consists of internal as well as leaf nodes.) , $T(S)$ is the minimial steiner subtree of T which connects the vertices in S. Moreover, 2-degree vertices in this steiner tree are eliminated replacing two edges with a new edge with weight of the new edge being sum of the weights of the previous two.*

Theorem 1. *TOC is NP-hard.*

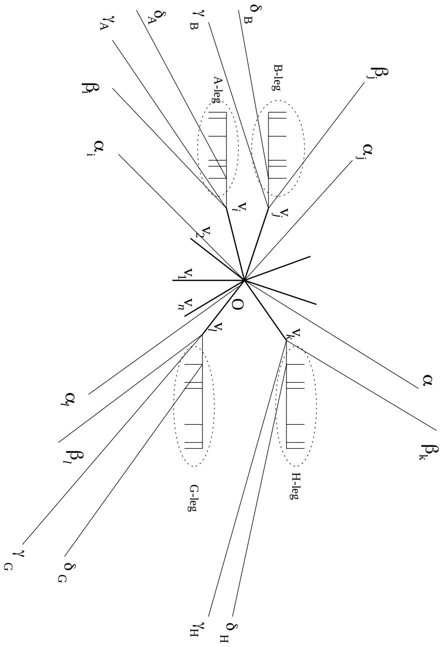


Fig. 5. Construction

Proof : The reduction is from UQC problem. Given an instance of UQC, we show how to construct an instance of TOC. Let $I = (S, Q)$ be an instance of UQC. $|S| = n, |Q| = m$. From I we construct a midpath tree T_{Δ} which would uniquely represent the triangle order Δ. We start with a star on n leaves in $S = \{v_1, v_2, ..., v_n\}$. Assign unit weight to each of these edges. Let the center of these star be o. Attach $2n$ more leaves α_i, β_i with $2n$ new edges of the form $o\alpha_i$

and $v_i\beta_i$ for all $i \in \{1, .., n\}$. Let $W(o\alpha_i) = W(v_i\beta_i) = 200000nm + 8ni$ for each i. α_i, β_i are dummy points.

Now, for each quartet $q_i = (v_{i_1}v_{i_2}, v_{i_3}v_{i_4})$ construct a weighted tree T_i as in the figure for generalized counter example. All of these T_i's have the same structure but they differ in weights according to parameter i. Now, take the four legs $A_i - leg, B_i - leg, G_i - leg, H_i - leg$ and attach these with the edge of weight $z - 1$ to vertices $v_{i_1}, v_{i_2}, v_{i_3}, v_{i_4}$ as in figure []. At the endpoints of each of these attachment edges attach two more leaves (these are also dummy points) $\gamma_{A_i}, \delta_{A_i}$ or $\gamma_{B_i}, \delta_{B_i}$ or $\gamma_{G_i}, \delta_{G_i}$ or $\gamma_{H_i}, \delta_{H_i}$ depending on the *leg* being attached.
$W(v_{i_1}, \gamma_{A_i}) = W(\hat{r}_i, \delta_{A_i}) = 30000nm + 400mi + 40n$
$W(v_{i_2}, \gamma_{B_i}) = W(\bar{r}_i, \delta_{B_i}) = 30000nm + 400mi + 80n$
$W(v_{i_3}, \gamma_{G_i}) = W(\hat{q}_i, \delta_{G_i}) = 30000nm + 400mi + 120n$
$W(v_{i_4}, \gamma_{H_i}) = W(\bar{r}_i, \delta_{H_i}) = 30000nm + 400mi + 160n$
Once we have done this for all the quartets, the resulting weighted tree is a super-imposition of T_i's with their respective edges o_ip_i contracted to a single vertex o. And also they share edges ov_j's according to quartet constraints. Next we define, mid-points over this tree $T^{'}$. For any leaves a, b belonging to same T_i the midpoint $M_{T_i}(a, b)$ is the edge on which the midpoint of weighted path $P_{T_i}(a, b)$ falls. The midpoint $M(a, b)$ is the edge in $T^{'}$ corresponding to $M_{T_i}(a, b)$. If a, b belong to T_i, T_j respectively with $i < j$ then $M(a, b)$ is defined according to weights in $T^{'}$. Note that this means that if a, b are not dummy points,the midpoint will be either of the edges $\hat{u}_j\hat{a}_j, \bar{u}_j\hat{b}_j, \hat{m}_j\hat{g}_j, \bar{m}_j\hat{h}_j$ depending on which *leg* of T_j b belongs. Now, observe that each non-leaf edge of $T^{'}$ has some midpoint falling on it. So, T_Δ is infact same as $T^{'}$ along with the midpoint function M. The triangle order Δ is as defined by this midpath tree T_Δ.

Now, the question is: is this midpath tree expandable? i.e. is this triangle order Δ realizable?

The following two lemmas would complete the proof of the theorem. □

Lemma 7. *If the triangle order Δ is realizable then answer to UQC question is Yes.*

Proof : Let T be the tree which realizes Δ. T is an expansion of T_Δ. Consider the constraints imposed by each T_i for all $i \in \{1, .., m\}$. These imply that there is an edge e such that $\hat{A}_i, \hat{B}_i$ are on one side of e and $\hat{G}_i, \hat{H}_i$ are on the other. Since o is the only point where such an expansion is possible, this edge e would also separate v_{i_1}, v_{i_2} from v_{i_3}, v_{i_4}. Because This means all the quartet constraints are satisfied in $T(S)$. □

Lemma 8. *If the answer to UQC question is Yes, then Δ is realizable.*

Proof : Consider T_Δ constructed as above. $T_\Delta(S)$ is a star with vertex o in the center and vertices $v_1, v_2, .., v_n$ as leaves. Now, since the answer to UQC question is Yes, consider the expansion of this star which provides solution to UQC problem. Assign unit weight to each of the newly expanded edges as well as the leaf edges. Note that there are atmost n new edges introduced since we are expanding a star on n leaves. Now consider the correspoding expansion in T_Δ, call this tree T. We show that T ,indeed, is the tree which realizes Δ. The rest part of T comes from different T_i's corresponding to each quartet. For weight function on those edges consider, $T(V(T_i))$. Figure [] shows the sketch of $T(V(T_i))$. There is unique edge, say e_i, separating $\hat{A}_i, \hat{B}_i$ from $\hat{G}_i, \hat{H}_i$. Set the parameter w to weight of e_i (instead of unit) in the corresponding *leg*'s. Adjust the weights of $v_{i_1}\hat{r}_i, v_{i_1}\bar{r}_i, v_{i_3}\hat{q}_i, v_{i_4}\bar{q}_i$ so that distances of $\hat{r}, \bar{r}, \hat{q}, \bar{q}$ from edge e_i equals $z(= 4n)$.

This weighted tree T indeed represents the same triangle order Δ. This can be verified by checking that all the midpoints remain on the same edges, giving the same triangle order. Note that no midpoint falls on the newly expanded parts. For the midpoints of the pair of leaves belonging to the same T_i, they still remain on the same edges giving the same order, since as noted in section 3, the value of parameter w is within the range of 1 and n. The midpoints among the pair of dummy points remain at the stem of the dummy point which had larger stem earlier. The midpoints of among the pair of dummy points with equal stems, originally, stay on the same edges. The midpoints of non-dummy points belonging to two different T_i and T_j, $i < j$, remain on the same edge which is one of $\hat{u}_j\hat{a}_j, \bar{u}_j\hat{b}_j, \hat{m}_j\hat{g}_j, \bar{m}_j\hat{h}_j$.

Hence, T realizes Δ. □

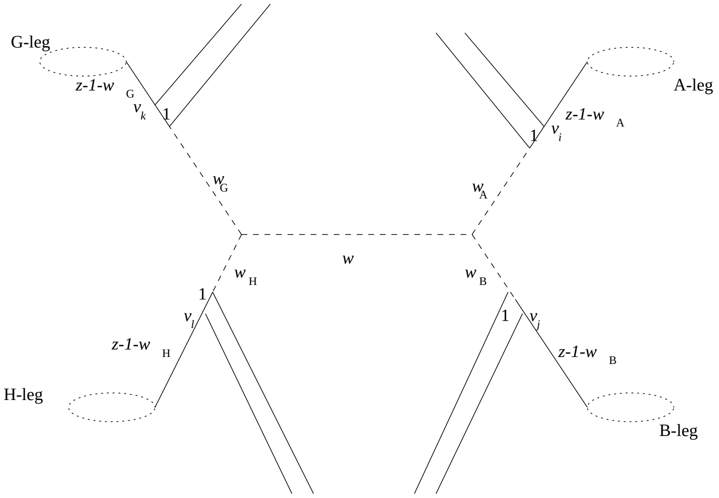


Fig. 6. UQC implies TOC

5 NP-hardness of Total Ordinal Clustering (OC)

In this section we show that given a total order τ on pairwise distances it is NP-hard to determine whether there exists a tree T which realizes τ. Again as in the previous section the reduction is from the UQC problem. Given an instance of UQC, we follow exactly the same construction here, except that the weights involved are slightly different. The parameter z, which was same $(4n)$ for all T_i's in TOC, is different for different T_i's. So we call it z_i. This is done to force the strict total order on non-critical (not involved in counter example table) distance pairs. We choose z_i's for $1 \leq i \leq m$ such that for $0 \leq p, q, p', q' \leq 400$ we get $|(pz_i + qz_j) - (p'z_{i'} + q'z_{j'})| \geq 1000n$ for any i, j, i', j', unless $i = i', j = j', p = p', q = q'$. These numbers are the the so called k-weighted well-separated numbers each multiplied by the factor of $1000n$ and $k = 400$ here. Let z_m be the highest of these z_i's. Then, we define z'_i for $1 \leq i \leq 8m + n$ as the (unweighted) well separated numbers each multiplied by $2z_m$. Now, for each T_i we choose $x_i = 50z_i$ and $y_i = 100z_i$. Each of the distance,

$\hat{m}\hat{M} = \hat{n}\hat{N} = \hat{q}\hat{Q} = z'_{4i-3}$	$\bar{m}\bar{M} = \bar{n}\bar{N} = \bar{q}\bar{Q} = z'_{4i-2}$
$\hat{r}\hat{R} = \hat{s}\hat{S} = \hat{t}\hat{T} = \hat{u}\hat{U} = z'_{4i-1}$	$\bar{r}\bar{R} = \bar{s}\bar{S} = \bar{t}\bar{T} = \bar{u}\bar{U} = z'_{4i}$

While constructing T', which is the weighted tree, we make the following weight changes, $W(o\alpha_i) = W(v_i\beta_i) = z^{'}_{8m+i}$
$W(v_{i_1}, \gamma_{A_i}) = W(\hat{r}_i, \delta_{A_i}) = z^{'}_{4m+4i-3}$
$W(v_{i_2}, \gamma_{B_i}) = W(\bar{r}_i, \delta_{B_i}) = z^{'}_{4m+4i-2}$
$W(v_{i_3}, \gamma_{G_i}) = W(\hat{q}_i, \delta_{G_i}) = z^{'}_{4m+4i-1}$
$W(v_{i_4}, \gamma_{H_i}) = W(\bar{r}_i, \delta_{H_i}) = z^{'}_{4m+4i}$

Apart from these weight changes the construction is exactly the same. Now, any pairwise distance in T' consists of weighted sum of atmost two z_i's and atmost two z'_i's. In any T_i, the distance of each point from the center edge o_ip_i, differs in the weight of z_i. Hence, each distance in T', which has two end point in different T_i's, has unique composition in terms of z_i's,z'_i's and their respective weights. The well-separation property guarantees that any pair of distances, if it doesn't have all 4 end points in the same T_i, differs by atleast $500n$. This is because the component of the distances due to w_i's as well as constant part is much smaller than the one due to z_i's and z'_i's.

Now, to define the total order we first give the pairwise distance values and then the total order will be defined by these values. For any two leaves v_p, v_q belonging to the same T_i, the distance $dist(v_p, v_q) = d_{T_i}(v_p, v_q)$ and if they belong to different T'_is then $dist(v_p, v_q) = d_{T'}(v_p, v_q)$. Again we define, midpoints on T'. For any leaves v_p, v_q belonging to same T_i the midpoint $M_{T_i}(v_p, v_q)$ is the edge on which the midpoint of weighted path $P_{T_i}(v_p, v_q)$ falls. $M(v_p, v_q)$ is the edge in T' corresponding to $M_{T_i}(v_p, v_q)$. If v_p, v_q belong to different T'_is them $M(v_p, v_q)$ is the midpoint of the weighted path $P_{T'}(v_p, v_q)$ in T'. Note that T' taken as unweighted tree is indeed the midpath tree T_τ with midpath function defiend as above. Also, because of the well-separation property, no midpoint is within the distance $100n$ from the center point o in T'.

The proof of lemma 1 is applicable as it is over here. So if the total order is realizable then the answer to the UQC question is Yes.

For proving the other way round again we do the same construction as in the proof of lemma 2. If the answer to the UQC question is Yes, then we expand the center vertex o according to the UQC soltion and assign unit weight to each of these newly expanded edge. Then, for each T_i take w as the number of edges that separeate vertices v_{i_1}, v_{i_2} from v_{i_3}, v_{i_4}. Note that this will also change the corresponding weights in T'. Now, no distance in T' changes by more than $10n$ (since $w \leq n$) by this expansion and correspoding increments in w values. So, due to well-separatedness, the same order is maintained for the distance pairs which do not consists of four leaves in the same T_i. For distance pair within the same T_i, whether $w = 1$ or w is number between 1 and n maintains the same order. Also, the order of distances involving $\alpha_i, \beta_i, \delta_A, \gamma_A, ...$ (dummy points not belonging to any T_i) remains unaltered during the expansion. Hence this expanded weighted tree T, represents the desired total order.

Theorem 2. *Total Ordinal Clustring (OC) is NP-hard.*

□

6 Ordinal Clustering in other Metric Spaces

In this paper we considered the problem of embedding orders into trees. There are other related metric spaces closely related to trees, where embedding orders could be interesting problem. For example on path (or line), the problem can be solved in polynomial time by mid-path tree algorithm (note that the path is subcase of trees). However, if it is a partial order i.e. incompletely specified orders then again the problem on path can be shown to be NP-complete by reduction to betweenness problem. In euclidean space, this problem could be determined by semi-definite programming. Also orders could be embedded into l_∞ metric space, because any distance metric can be isometrically embedded into l_∞. It remains interesting question to determine the lowest dimension of the target space (euclidean or l_∞) required to embed the order. In this sense we could define a property called dimensionality of an order.

7 Conclusion and Future Work

Our result, in a way, ends the quest for constructing weighted phylogeny from orders on a negative note (unless $P = NP$). There are some approximation criteria which could be considered. For example, dropping out minimum number of leaves so that order becomes embeddable or finding a tree embedding with the least number of inversion pairs in the order. All these criteria would become NP-hard, but one could seek some approxiamtions there.

Also, this is first of kind NP-hardness result known in phylogeny construction where the input data is fully specified. Most similar NP-hardness results known were for tree construction from incomplete distance matrix or erroneous data or incompeletely specified orders. Also, this is the first result when weighted phylogenies are addressed for representing orders.

References

[1] S. Kannan and T. Warnow, *Tree reconstruction from partial orders*, SIAM Journal of Computing, vol 24, no3., pp. 511-519, June 1995.

[2] P. Kearney , R. Hayward and H. Meijer, *Phylogenies from relative dissimilarity*, Algorithmica: Special issue on computational biology, 25, pp. 196-221, 1999.

[3] S. Kannan, *personal communication.*

[4] D. Robak, *unpublished document, personal communication.*

[5] R. Shah and M. farach-Colton, *On the Midpath Tree Conjecture: A Counter-example*, in proceedings of Symposium on Discrete Algorithms (SODA), 2001.

[6] P. Kearney, *A six-point condition for ordinal matrices*, Manuscript, 1995.

[7] William H. E. Day, *Inferring phylogenies from dissimilarity matrices*, Bull. of Mathematical Biology, 49:4, 1987.

[8] M. Farach, S. kannan and T. Warnow, *A robust model for finding evolutionary trees*, Algorithmica, 13:1, 1995.

[9] J. C. Culberson and P. Rudnicki, *A fast algorithm for constructing trees from distance matrices*, Information Processing Letters, 30, 1989.

[10] J. Hein, *An optimal algorithm to reconstruct trees from additive matrices*, Bull. of Math. Bio., 51, 1989.

[11] M. A. Steel, *The complexity of reconstructing trees from qualitative characters and subtrees*, J. Classification, 9, 1992.

[12] M. S. Waterman, T. F. Smith, M. Singh, W. A. Bayer, *Additive evolutionary trees*, J. Theor. Biol., 64, 1977.

[13] P. Buneman, Mathematics in Archeolgical and Historical Sciences (F. R. Hobson, D. G. Kendall, P. Tautu eds), Univ Press, Edinburgh, pp 387-395.