On the Midpath Tree Conjecture: A Counter-Example

Rahul Shah* Martin Farach-Colton†

Introduction. Clustering data based on pairwise distances is a fundamental problem. Weighted trees can be used to represent hierarchical clusters. Multidimensional Scaling (MDS), in some formulations, is the problem of clustering data based on preserving their *relative* (rather than absolute) distances. We call this an *ordinal clustering*.

A particular instance of this problem has been considered in the algorithmic computational biology community [KW, KHM]: Given a (total or partial) order on the pairwise distances between points, give a weighted tree on those points so that the pairwise pathlengths between points in the tree satisfies this order. This is, therefore, simply the MDS problem for hierarchical clustering, and we will refer to it as the HMDS problem.

It has been conjectured [Kan] that there is a polynomial time algorithm to solve the HMDS problem, both while preserving the total order of the points, as well as while relaxing the order to certain types of partial orders. This algorithm is called the *Midpath Tree Algorithm*. While it clearly runs in polynomial time, it was not known to always produce the correct output for HMDS. In this short paper, we show that it does not, in fact, always produce the correct output.

Previous work includes an algorithm described by Kearney et al in [KHM] to construct an *unweighted* tree, if it exists, which realizes the total order on pairwise distances. Clearly, in many cases an unweighted tree may not exist for an order, while a weighted tree does exist, so solving the unweighted case sheds little light on the weighted HMDS problem. Kannan and Warnow [KW] solved a similar problem to realize certain types of partial orders and posed the weighted case as an open problem. They specifically worked with a partial order constructed from triplets of data points which we call a *triangle order*. A triangle order is a partial order on distances so that that distances within each triplet of points are totally ordered.

Before presenting the Midpath Tree Algorithm and its counter-example, we present some preliminary observations.

Realizability and LP's. Define d_{T_w} to be the distance metric of tree T under a non-negative weight function w, where $d_{T_w}(s,t)$ is sum of weights of 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 $d_{T_w}(a,b) < d_{T_w}(c,d)$ whenever $d(a,b) <_P d(c,d)$[1].

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

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

$$x \geq 0 \quad (2)$$

where $\delta > 0$ is a constant.

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. 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'.

Midpath Trees. 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 $m : S \times S \to E(T)$ which maps each pair of leaves a, b to the edge $m(a,b)$ 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 midpoints falling on them to obtain an unweighted tree which we call the *midpath tree* T_Δ. The *midpath function* m for T_Δ is borrowed from T. For any pair $a, b \in S$, let $T_\Delta^{a/b}$ and $T_\Delta^{b/a}$ be the connected components of $T_\Delta - m(a,b)$ containing a and b respectively. Then, $c \in T_\Delta^{a/b} \Leftrightarrow d(a,c) <_\Delta d(b,c)$. The midpoint edge $m(a,b)$ gives a bipartition of points in S based on whether they are closer to a or to b. 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 bipartition property can be defined.

Clearly, the existence of the midpath tree is a prerequisite for the existence of a tree T which realizes the

*Dept. of CS, Rutgers University, sharahul@paul.rutgers.edu
†Google Inc., martin@google.com

[1] For the sake of simplicity and conciseness, we shall only consider orders with strict inequalities

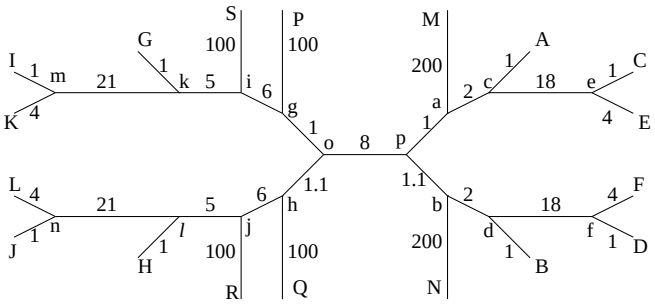

Figure 1: Δ not Realizable as Midpath Tree

triangle (or total) order and any such T is an expansion of T_Δ [KHM]. Intuitively, the existence of midpath tree indicates that the triangle (or total) order has a tree-like nature. Trivially [KHM] if the midpath tree exists for an order, it is unique. It can be obtained by starting with a *star* tree and expanding it until all the midpoints exist. This runs in $O(n^3)$ time. However, an optimal $O(n^2)$ algorithm for constructing the midpath tree from a triangle (or total) is known [KHM]. So, the main issue is: Is it always possible to define some weight function on T_Δ to realize Δ or do we need to expand T_Δ? In the latter case, how?

The Midpath Tree Algorithm. The midpath tree algorithm to realize a triangle (or total) order Δ is as follows: Construct the midpath tree T_Δ. Check whether or not there is some weight function on the edges of T_Δ which realizes Δ, by checking if linear constraints (1) and (2) are feasible. If feasible, the algorithm provides a realization of Δ. Otherwise, the algorithm claims that Δ is *not realizable*.

This algorithm clearly runs in polynomial time. However, its correctness is based on the following conjecture [Kan],[KHM].

The Midpath Tree Conjecture. *If a triangle order Δ is realizable as some tree, then it is realizable as the midpath tree T_Δ.*

An example tree consisting of 6 leaves, disproving the similar conjecture for total orders was known, as was an example showing that some total orders are not realizable at all, despite the existence of the midpath tree [Rob]. In the next section, we give a counter-example to the midpath tree conjecture, though still leaving open the problem of finding the correct algorithm for realizing the triangle order.

Counter-Example.

Theorem 1 *The midpath tree conjecture is false.*

Proof : Figure 1 gives the counter-example. 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,

$$pd + pa + oj + og < oc + oi + ob + oh^2$$

By a similar analysis, we can show that

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

Summing yields $op > 0$. Thus, the midpath tree can not 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	$HQ <_\Delta QA$	$oH - 2oh < oA$

Remarks and Future Work. Future work is mainly to find an algorithm that generates an expansion of the midpath tree which would realize Δ. To do this, we might want to pick a vertex which needs expansion, then look at a minimal set of constraints which show infeasibility and involve this vertex, and check whether they indicate some expansion of this vertex in order to achieve feasibility. We hope that the tools used to generate this counter-example will provide an insight into techniques to expand the midpath tree.

References

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

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

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

[Rob] D. Robak, *personal communication.*

[2] these are tree distances under any w