On the Placement of Euclidean Trees

Sue Whitesides & Rongyao Zhao

School of Computer Science, McGill University, 3480 University Street, Montreal, Quebec, Canada H3A 2A7

Abstract

A *Euclidean tree* is a tree whose edges are assigned positive integer lengths. A *placement* of a Euclidean tree is an assignment f of its vertices to points in space such that for each tree edge (u,v), the Euclidean distance between $f(u)$ and $f(v)$ is equal to the length assigned to the edge. We define the EUCLIDEAN TREE PLACEMENT problem as follows: given a Euclidean tree together with the desired integer coordinate locations for its leaves, determine whether the tree has a placement that assigns the leaves to the specified locations. In this paper, we prove that EUCLIDEAN TREE PLACEMENT is NP-hard.

Our interest in tree placement problems arose in the context of designing a computer graphics program in which the user designs a jointed object and places it in a scene subject to constraints. For example, if the object is the model of a human figure climbing a ladder, the user might give locations for the hands and feet and request that the rest of the body be placed by the program. The program must then determine a configuration (if one exists) that satisfies the user's constraints. Problems of this type also arise in choosing grasp positions for robot hands and in studying the conformation of macromolecules in molecular physics.

The basic idea of the proof is to use a reduction from SET PARTITION. The location constraints on the leaves of the tree constructed by the reduction ensure that a chain of internal tree vertices are effectively confined to a large circle and that the two end vertices of the chain coincide. Edges joining vertices on the circle subtend arcs that ideally should have arc lengths equal to the weights from the partition problem. Then the weights could be partitioned if and only if the corresponding arc lengths sum to 0, where the arc lengths are signed + or – according to which set the weights belong. Because the tree edges have integer lengths and because the trigonometric computations needed to calculate their lengths cannot be done with infinite precision, certain errors are introduced. These are compensated for by the addition of a "gadget" consisting of extra tree edges. An interesting aspect of the proof is to use a "gap" technique to show that the approximations can be done well enough to give an exact answer in polynomial time.