Proceedings of International Conference
on Algorithms

An Approximate Algorithm for Evolutionary Tree Reconstruction from Distance Matrix

B.Y. Wu & C.Y. Tang
Dept. of Computer Science, National Tsing Hua University,
Hsinchu, Taiwan, R.O.C.
E-mail: dr838305,cytang@cs.nthu.edu.tw

Abstract:

Most optimization problems for evolutionary tree reconstruction from distance matrix are either NP-complete or still unknown. In this paper, we present an approximate algorithm under an important optimization criteria: minimum tree size. Our algorithm runs in $O(n^3)$ time and constructs a tree which size is no more than 3 times of the optimal.

Keywords: approximate algorithms, evolutionary trees

1. Introduction

Constructing evolutionary trees (phylogenetic trees) from distance matrix is a classical problem in computational biology. Given the pairwise dissimilarities (distances) between species, people hope to reconstruct a tree to find the evolutionary relations between species. An evolutionary tree is an edge-weighted tree with leaves as species. On an evolutionary tree T, the distance between two species, denoted by d(T,i,j), is the total weight of the path of the two leaves. A distance matrix M is a symmetry square matrix, and M[i,j] is the dissimilarity between species i and j. M is said to be additive if there is an evolutionary tree T realizing it, that is, d(T,i,j)=M[i,j] for all i and j. If the distance data are additive, there are efficient algorithms for reconstructing the tree[1,4,6,7], and the uniqueness of the tree had been shown [7]. But in most of the cases, the distance matrix is not additive. Scientists try to construct an optimal tree T under some optimization criteria, and such that $d(T,i,j)\geq M[i,j]$ for all i and j. The reason why the distances should be no less than the given ones is that the distances obtained by sequences alignment were believed as low bounds [2,7]. In [7], explanations of this phenomena were also given.

However, the reconstruction problems under all popular criteria are either known or conjectured to be NP-complete [5]. In [2], three criteria were considered. They are L^1-norm, L^∞-norm, and minimum tree size (MTS for brief). The problems was defined to construct tree T such that $d(T,i,j) \geq M[i,j]$ for all i and j, and such that $\sum_{i,j}\{d(T,i,j)-M[i,j]\}$, $\max_{i,j}\{d(T,i,j)-M[i,j]\}$, or $\sum_{e \in T} w(e)$ is minimized respectively. Reconstruction of ultrametric tree is also considered in the paper. An evolutionary tree is ultrametric if every leaf has the same distance from the root. In [2], an polynomial time algorithm for constructing ultrametric tree under L^{∞}-norm was developed. However, for the other optimization problems of tree reconstruction, they are all NP-complete or still open [2,6]. (see Table 1) For the minimum size problem of ultrametric tree, it had been shown that there is a constant $\varepsilon > 0$ such that no polynomial time algorithm can approximate the optimal solution within a ratio of n^{ε} unless NP=P. For the other problems, no approximate algorithm has been presented.

In this paper, we consider reconstruction problem under the minimum tree size criterion (MTS problem). The MTS problem is to reconstruct an evolutionary tree T such that $\sum_{e \in T} w(e)$, is minimized. An approximate algorithm is presented in this paper. The algorithm can construct an evolutionary tree with size no more than 3 times of the optimal in $O(n^3)$ time complexity.

The remaining paragraphs are organized as follows: We define some notations in Section 2. In Section 3, the algorithm for MTS is presented. In Section 4, a conclusion is given.

2. Definitions & notations

Definition: [2] A phylogenetic tree (or evolutionary tree) for a species set S is a rooted tree in which the leaves are labeled by the species in S.

Definition: [7] A metric tree (unrooted) T is a tree in which each edge has a nonnegative weight and each internal node has degree 3. Let i,j be two nodes of tree T, d(T,i,j) denotes the distance

Table 1,

Desired tree	L^1-norm	L^{∞}-norm	minimum tree size
Optimal Ultrametric	NPC [2]	$\Theta(e+n\log n)$ [2]	not-approx [2]
Optimal Additive	NPC [2]	open	open
Approximate additive			Section 3

between I and j, that is, the total edge weight of the unique path between i and j on T.

Since any tree with nonnegative weights on edges can be transformed to a metric tree without changing the distances between leaves, we shall assume an evolutionary tree is a metric tree in the remaining paragraph.

Definition: [2] For a set with n species, a *distance matrix (or dissimilarity matrix)* is an n×n matrix in which each element represents the distance between the two species. Let M be a distance matrix, it satisfies the following conditions: (1) M[i,i]=0, (2) M[i,j]=M[j,i], (3)M[i,j]>0, (4)M[i,j]≤M[i,k]+M[k,j], for all i,j, and k.

Definition: [7] A distance matrix M of species set V is said to be additive if there is an evolutionary tree T with leaf set V realizing M, that is M[i,j]=d(T,i,j) for all leaves i and j.

Problem MTS:(Minimum Tree Size Problem)

Given a distance matrix M, find a metric tree T, such that d(T,i,j) ≥M[i,j] and the size of T , defined by SIZE(T)=$\sum_{e\in T}$w(e), is minimized. We shall use MTS(M) to denote the solution of the problem.

Definition: Let M be an n×n distance matrix, M is corresponding to a undirected complete graph G=<V,E> where V={$v_1,v_2,...,v_n$} and we)=M[i,j] for any edge e=(v_i,v_j). A Hamiltonian tour on G is a circle starting from any node and visiting each node exactly once. TSP(M) denotes the minimum length of a Hamiltonian tour on G, which is just the solution of Traveling Salesperson Problem of G.

3. Minimum tree size problem

We first show a low bound of MTS problem.

Lemma 3.1: Let M be an additive matrix and T be the tree realizing it, then TSP(M)=2×SIZE(T).

proof: Since M is additive, M[i,j]=d(T,i,j). Starting from a leaf of T, if we travel on T to visit every leaf and then back to the starting leaf, we must visit each edge at least twice. Since such a traveling sequence corresponds to a tour on M, TSP(M)≥2×SIZE(T). It is easy to show that if the traveling sequence is the Depth-First-Search sequence then each edge will be visited exactly twice. Since TSP(M) is the minimum length of tour, TSP(M)=2×SIZE(T) and completes the proof.

□

Lemma 3.2: If T=MTS(M) then SIZE(T)≥TSP(M)/2.

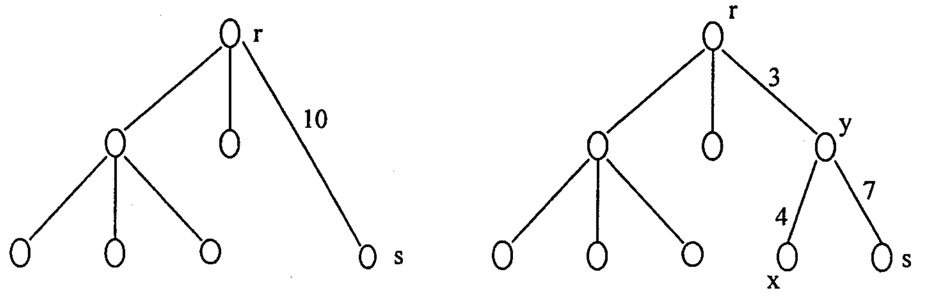

Fig.1: A solution ((r,s),4,3) of insertion problem: (r,s) is the attached edge and splitted into (r,y), (y,s), and a new edge (y,x) is inserted.

proof: Let A be the corresponding additive matrix of T, that is, A[i,j]=d(T,i,j). From Lemma 3.1, TSP(A)=2×SIZE(T). Since A[i,j]≥M[i,j] for all I and j, TSP(A)≥TSP(M). So, TSP(M)≤2×SIZE(T).

□

Since a distance matrix satisfies the triangle inequality, there is a polynomial time algorithm to find an approximate solution of the Traveling Salesperson Problem.

Lemma 3.3: Given a distance matrix, an approximate solution ATSP(M) can be found in O(n^3) time and ATSP(M)≤1.5×TSP(M). [3]

Before presenting the approximate algorithm, we state a insertion problem and give an algorithm for it.

Definition: Min-Size-Insertion Problem:

Given an evolutionary tree T of leaf set {1..n} and a vector $D_x=(a_1,a_2,....,a_n)$ with $a_i+d(T,i,j)\geq a_j$ ∀ i and j, find a way to insert leaf x into T such that $d(T,x,j)\geq a_j$ ∀j and the size of the resulting tree is minimized. We shall use MSI(T,x) to denote the resulting tree.

We now show how to solve this problem. For any edge e=(r,s) of T, deleting e will result in two subtrees. We call the subtree with s as T_r, and the other as T_s. The leaf sets of T_r and T_s are V_r and V_s respectively. A solution of Min-Size-Insertion problem is a triple <e,h,k>. e=(r,s) is the attached edge and split into (r,y) and (y,s), k=w(r,y), and h=w(y,x) (Fig. 1). The algorithm for solving the Min-Size-Insertion problem is modified from [8], we discuss here briefly. We shall define some notations first.

Definition: For each edge e=(r,s), $\beta_i=d(T,s,i)$ if leaf i∈ V_s and $\beta_i=d(T,r,i)$ if leaf i∈ V_r. $C_e(T,D_x)=\min\{h$: <e,h,k> is a solution}. $C(T,D_x)=\min\{C_e(T,D_x)$: for all e of T}. $p_{max}(e,s)=\max\{a_i-\beta_i : \forall\ i\in V_s\}$. $p_{max}(e,r)=\max\{a_i-\beta_i : \forall\ i\in V_r\}$.

Lemma 3.4: The solution of Min-Size-Insertion problem can be found in O(n) time.

proof: If $C_e(T,D_x)$ can be found in O(n) time for all edge, the problem can also be solved in O(n) time by finding the minimum among $C_e(T,D_x)$. We shall first show how to find $C_e(T,D_x)$ for any edge.

Observe that any feasible solution <e,h,k> is to split e=(s,r) into (s,y), (y,r) and insert an edge (y,x). (y may coincide with s or r).

The problem is to select h and k with minimum h subject to $w(e)\geq k\geq 0$, $h\geq 0$, and $d(T,x,i)\geq a_i\ \forall i$.

Since

$$d(T,x,i)=\begin{cases} h+k+\beta_i & \text{if } i\in V_r \\ h+w(e)-k+\beta_i & \text{if } i\in V_s \end{cases}$$

, the conditions can be rewriten as

$$\begin{cases} h+k\geq p_{\max}(e,r) \\ h+w(e)-k\geq p_{\max}(e,s) \\ w(e)\geq k\geq 0 \\ h\geq 0 \end{cases}$$

It can be shown that the solution of this linear programming is $h=\max\{0,p_{\max}(e,r)-w(e), p_{\max}(e,s)-w(e), (p_{\max}(e,r)+p_{\max}(e,s)-w(e))/2\}$. If $p_{\max}(e,r)$ and $p_{\max}(e,s)$ are known, the solution can be found in constant time. In [8], there is an algorithm for finding $p_{\max}(e,r)$ and $p_{\max}(e,s)$ for all edges in O(n) time. So, the whole problem can be solved in O(n) time.

□

We present the approximate algorithm in Figure 2.

Lemma 3.5: $SIZE(T_i)\leq SIZE(T_{i-1})+M[i-1,i]\ \forall\ 2\leq i\leq n$.

proof: Let e=(r,i) be an edge of T where r is an internal node, we claim that replacing i with r_1 and inserting (r_1,i), $(r_1,i+1)$ such that $w(r_1,i-1)=0$ and $w(r_1,i)=M[i-1,i]$ is a feasible solution. To see the validity, for any $j\leq i-1$, $d(T,j,i)=w(r_1,i)+w(r_1,j)$

```
Algorithm APX_MTS

Input: a distance matrix M and its corresponding complete graph G.

Output: an evolutionary tree T.
Step1: Find the approximate solution ATSP(M) and its corresponding sequence S. We assume
        S=<1,2,...,n>
Step2: Set $T_0$ as an empty tree.
        For i=1 to n do
            $T_i$=MSI($T_{i-1}$,i)

Step3: Output $T_n$ as an approximate solution of MTS problem.
```

Fig. 2

$=M[i-1,i]+d(T,j,i-1)\geq M[i-1,i]+M[i-1,j]\geq M[i,j]$.

Since MSI(T,i) takes the minimum among all feasible solutions, so,

$SIZE(T_i)\leq SIZE(T_{i-1})+M[i-1,i]\ \forall\ 2\leq i\leq n$

□

From Lemma 3.5, it is ease to show the following corollary.

Corollary 3.1: $SIZE(T_n)\leq ATSP(M)$.

Theorem 1: Algorithm APX_MTS reconstructs an evolutionary tree in $O(n^3)$ time with size no more than 3 times of the optimal.

Proof: Since $SIZE(MTS(M))\geq TSP(M)/2$ (Lemma 3.2) and

$TSP(M)\geq ATSP(M)/1.5$, (Lemma 3.3)

$ATSP(M)\leq 3\times SIZE(MTS(M))$. From Corollary 3.1, APX_MTS constructs an evolutionary tree with size no more than ATSP(M). The time complexity for finding ATSP(M) is $O(n^3)$, and for inserting leaves is $O(n^2)$. This results time complexity $O(n^3)$ for the whole algorithm.

□

4. Conclusion

In this paper, we examine the problem for reconstructing an evolutionary tree from distance matrix. An approximate algorithm were developed for the minimum tree size criterion. The algorithms can find solution with performance ratio 3 and polynomial time $O(n^3)$. There are still open problems for evolutionary tree reconstruction. The complexity of minimum tree size problem is still unknown. Another important optimization criterion is L^1-norm. For this criterion, both the complexity and if there is approximate algorithm are still open.

References:

[1]: J. Culberson and P. Rudnicki, A fast algorithm for constructing trees from distance matrices, *Inform. Process. Lett.*, 30:215-220,1989.

[2]: M. Farach, S. Kannan, and T. Warnow, A robust model for finding optimal evolutionary tree, *Algorithmica*, 13:155-179, 1995.

[3]: M.R. Garey and D.S. Johnson, *Computers and Intractablity: Aguide to the Theory of NP-Completeness* (Freeman, San Francisco, CA, 1979).

[4]: S. Kannan, E. Lawler, T. Warnow, Determining the evolutionary tree, *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pp.475-484, San Francisco, CA, Jan. 1990.

[5]: S. Kannan and T. Warnow, Tree reconstruction from partial orders, *SIAM J. Computing*, vol. 24, No. 3, pp.511-519, 1995.

[6]: M. Krivanek, The complexity of ultrametric partitions on graph, *Inform. Process. Lett.*, 27(5):265-270, 1988.

[7]: M.S. Waterman, T.F. Smith, M. Singh, and W.A. Beyer, Additive evolutionary tree, *J. theor. Biol.*, 64:199-213, 1977.

[8]: B.Y. Wu and C.Y. Tang, Optimal algorithms for positioning unknown species in an evolutionary tree, manuscript.