

is at most  $M$ . The *circuit clustering problem* is to compute a feasible clustering  $\Gamma$  of  $G$  such that the delay of  $\Gamma$  is minimum among all feasible clusterings of  $G$ .

An early work of Lawler et al. [2] presented a polynomial-time optimal algorithm for the circuit clustering problem in the special case where all the gate delays are zero (i. e.,  $\delta(v) = 0$  for all  $v$ ).

### Key Results

Rajaraman and Wong [5] presented an optimal polynomial-time algorithm for the circuit clustering problem under the general delay model.

**Theorem 1** *There exists an algorithm that computes an optimal clustering for the circuit clustering problem in  $O(n^2 \log n + nm)$  time, where  $n$  and  $m$  are the vertices and edges, respectively, of the given combinational network.*

This result can be extended to compute optimal clusterings under any monotone clustering constraint. A clustering constraint is monotone if any connected subset of nodes in a feasible cluster is also monotone [2].

**Theorem 2** *The circuit clustering problem can be solved optimally under any monotone clustering constraint in time polynomial in the size of the circuit.*

### Applications

Circuit partitioning/clustering is an important component of very large scale integration design. One application of the circuit clustering problem formulated above is to implement a circuit on multiple field programmable gate array chips. The work of Rajaraman and Wong focused on clustering combinational circuits to minimize delay under area constraints. Related studies have considered other important constraints, such as pin constraints [1] and a combination of area and pin constraints [6]. Further work has also included clustering sequential circuits (as opposed to combinational circuits) with the objective of minimizing the clock period [4].

### Experimental Results

Rajaraman and Wong reported experimental results on five ISCAS (International Symposium on Circuits and Systems) circuits. The number of nodes in these circuits ranged from 196 to 913. They reported the maximum delay of the clusterings and running times of their algorithm on a Sun Sparc workstation.

### Cross References

► FPGA Technology Mapping

### Recommended Reading

1. Cong, J., Ding, Y.: An optimal technology mapping algorithm for delay optimization in lookup-table based fpga design. In: Proceedings of IEEE International Conference on Computer-Aided Design, 1992, pp. 48–53
2. Lawler, E.L., Levitt, K.N., Turner, J.: Module clustering to minimize delay in digital networks. IEEE Trans. Comput. **C-18**, 47–57 (1966)
3. Murgai, R., Brayton, R.K., Sangiovanni-Vincentelli, A.: On clustering for minimum delay/area. In: Proceedings of IEEE International Conference on Computer-Aided Design, 1991, pp. 6–9
4. Pan, P., Karandikar, A.K., Liu, C.L.: Optimal clock period clustering for sequential circuits with retiming. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **17**, 489–498 (1998)
5. Rajaraman, R., Wong, D.F.: Optimum clustering for delay minimization. IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst. **14**, 1490–1495 (1995)
6. Yang, H.H., Wong, D.F.: Circuit clustering for delay minimization under area and pinconstraints. IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst. **16**, 976–986 (1997)

## Phylogenetic Tree Construction from a Distance Matrix 1989; Hein

JESPER JANSSON

Ochanomizu University, Tokyo, Japan

### Keywords and Synonyms

Phylogenetic tree construction from a dissimilarity matrix

### Problem Definition

Let  $n$  be a positive integer. A *distance matrix of order  $n$*  (also called a *dissimilarity matrix of order  $n$* ) is a matrix  $D$  of size  $(n \times n)$  which satisfies: (1)  $D_{i,j} > 0$  for all  $i, j \in \{1, 2, \dots, n\}$  with  $i \neq j$ ; (2)  $D_{i,j} = 0$  for all  $i, j \in \{1, 2, \dots, n\}$  with  $i = j$ ; and (3)  $D_{i,j} = D_{j,i}$  for all  $i, j \in \{1, 2, \dots, n\}$ .

Below, all trees are assumed to be unrooted and edge-weighted. For any tree  $\mathcal{T}$ , the *distance* between two nodes  $u$  and  $v$  in  $\mathcal{T}$  is defined as the sum of the weights of all edges on the unique path in  $\mathcal{T}$  between  $u$  and  $v$ , and is denoted by  $d_{u,v}^{\mathcal{T}}$ . A tree  $\mathcal{T}$  is said to *realize* a given distance matrix  $D$  of order  $n$  if and only if it holds that  $\{1, 2, \dots, n\}$  is a subset of the nodes of  $\mathcal{T}$  and  $d_{i,j}^{\mathcal{T}} = D_{i,j}$  for all  $i, j \in \{1, 2, \dots, n\}$ . Finally, a distance matrix  $D$  is

called *additive* or *tree-realizable* if and only if there exists a tree which realizes  $D$ .

**Problem 1 (The Phylogenetic Tree from Distance Matrix Problem)**

INPUT: An distance matrix  $D$  of order  $n$ .

OUTPUT: A tree which realizes  $D$  and has the smallest possible number of nodes, if  $D$  is additive; otherwise, null.

See Fig. 1 for an example.

In the time complexities listed below, the time needed to input all of  $D$  is not included. Instead,  $O(1)$  is charged to the running time whenever an algorithm requests to know the value of any specified entry of  $D$ .

### Key Results

Several authors have independently shown how to solve The Phylogenetic Tree from Distance Matrix Problem. The fastest of these algorithms run in  $O(n^2)$  time<sup>1</sup>:

**Theorem 1 ([2,4,5,7,15])** *There exists an algorithm which solves The Phylogenetic Tree from Distance Matrix Problem in  $O(n^2)$  time.*

Although the algorithms are different, it can be proved that:

**Theorem 2 ([8,15])** *For any given distance matrix, the solution to The Phylogenetic Tree from Distance Matrix Problem is unique.*

Furthermore, the algorithms referred to in Theorem 1 have optimal running time since any algorithm for The Phylogenetic Tree from Distance Matrix Problem must in the worst case query all  $\Omega(n^2)$  entries of  $D$  to make sure that  $D$  is additive. However, if it is known in advance that the input distance matrix is additive then the time complexity improves, as shown by Hein [9]:

**Theorem 3 ([9,12])** *There exists an algorithm which solves The Phylogenetic Tree from Distance Matrix Problem restricted to additive distance matrices in  $O(kn \log_k n)$  time, where  $k$  is the maximum degree of the tree that realizes the input distance matrix<sup>2</sup>.*

The algorithm of Hein [9] starts with a tree containing just two nodes and then successively inserts each node  $i$  into the tree by repeatedly choosing a pair of existing nodes and computing where on the path between them that  $i$  should

be attached, until  $i$ 's position has been determined. (The same basic technique is used in the  $O(n^2)$ -time algorithm of Waterman et al. [15] referenced to by Theorem 1 above, but the algorithm of Hein selects paths which are more efficient at discriminating between the possible positions for  $i$ .)

The lower bound corresponding to Theorem 3 is given by:

**Theorem 4 ([10])** *The Phylogenetic Tree from Distance Matrix Problem restricted to additive distance matrices requires  $\Omega(kn \log_k n)$  queries to the distance matrix  $D$ , where  $k$  is the maximum degree of the tree that realizes  $D$ , even if restricted to trees in which all edge weights are equal to 1.*

Finally, note that the following special case is easily solvable in linear time:

**Theorem 5 ([5])** *There exists an  $O(n)$ -time algorithm which solves The Phylogenetic Tree from Distance Matrix Problem restricted to additive distance matrices for which the realizing tree contains two leaves only and has all edge weights equal to 1.*

### Applications

The main application of The Phylogenetic Tree from Distance Matrix Problem is in the construction of a tree (a so-called *phylogenetic tree*) that represents evolutionary relationships among a set of studied objects (e. g., species or other taxa, populations, proteins, genes, etc.). Here, it is assumed that the objects are indeed related according to a tree-like branching pattern caused by an evolutionary process and that their true pairwise evolutionary distances are proportional to the measured pairwise dissimilarities. See, e. g., [1,6,7,14,15] for examples and many references as well as discussions on how to estimate pairwise dissimilarities based on biological data. Other applications of The Phylogenetic Tree from Distance Matrix Problem can be found in psychology, for example to describe semantic memory organization [1], in comparative linguistics to infer the evolutionary history of a set of languages [11], or in the study of the filiation of manuscripts to trace how manuscript copies of a text (whose original version may have been lost) have evolved in order to identify discrepancies among them or to reconstruct the original text [1,3,13].

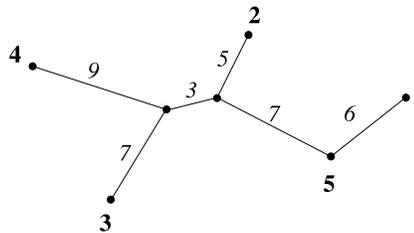
In general, real data seldom forms additive distance matrices [15]. Therefore, in practice, researchers consider optimization versions of The Phylogenetic Tree from Distance Matrix Problem which look for a tree that “almost” realizes  $D$ . Many alternative definitions of “almost” have been proposed, and numerous heuristics and approxima-

<sup>1</sup>See [5] for a short survey of older algorithms which do not run in  $O(n^2)$  time.

<sup>2</sup>For this case, the Culberson-Rudnicki algorithm [5] runs in  $O(n^{3/2} \sqrt{k})$  time for trees in which all edge weights are equal to 1, and not in  $O(kn \log_k n)$  time as claimed in [5]. See [12] for a counterexample to [5] and a correct analysis.

$$D = \begin{bmatrix} 0 & 18 & 23 & 25 & 6 \\ 18 & 0 & 15 & 17 & 12 \\ 23 & 15 & 0 & 16 & 17 \\ 25 & 17 & 16 & 0 & 19 \\ 6 & 12 & 17 & 19 & 0 \end{bmatrix}$$

a



b

### Phylogenetic Tree Construction from a Distance Matrix, Figure 1

**a** An additive distance matrix  $D$  of order 5. **b** A tree  $\mathcal{T}$  which realizes  $D$ . Here,  $\{1, 2, \dots, 5\}$  forms a subset of the nodes of  $\mathcal{T}$

tion algorithms have been developed. A comprehensive description of some of the most popular distance-based methods for phylogenetic reconstruction as well as more background information can be found in, e. g., Chapt. 11 of [6] or Chapt. 4 of [14]. See also [1] and [16] and the references therein.

### Cross References

- ▶ Distance-Based Phylogeny Reconstruction (Fast-Converging)
- ▶ Distance-Based Phylogeny Reconstruction (Optimal Radius)

### Acknowledgments

Supported in part by Kyushu University, JSPS (Japan Society for the Promotion of Science), and INRIA Lille – Nord Europe.

### Recommended Reading

1. Abdi, H.: Additive-tree representations. In: Dress, A., von Haeseler, A. (eds.) *Trees and Hierarchical Structures: Proceedings of a conference held at Bielefeld, FRG, Oct. 5–9th, 1987*. Lecture Notes in Biomathematics, vol. 84, pp. 43–59. Springer (1990)
2. Batagelj, V., Pisanski, T., Simões-Pereira, J.M.S.: An algorithm for tree-realizability of distance matrices. *Int. J. Comput. Math.* **34**, 171–176 (1990)
3. Bennett, C.H., Li, M., Ma, B.: Chain letters and evolutionary histories. *Sci. Am.* **288**, 76–81 (2003)
4. Boesch, F.T.: Properties of the distance matrix of a tree. *Quarterly Appl. Math.* **26**, 607–609 (1968)
5. Culberson, J.C., Rudnicki, P.: A fast algorithm for constructing trees from distance matrices. *Inf. Process. Lett.* **30**, 215–220 (1989)
6. Felsenstein, J.: *Inferring Phylogenies*. Sinauer Associates, Inc. (2004)
7. Gusfield, D.M.: *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, New York (1997)
8. Hakimi, S.L., Yau, S.S.: Distance matrix of a graph and its realizability. *Quarterly Appl. Math.* **22**, 305–317 (1964)
9. Hein, J.: An optimal algorithm to reconstruct trees from additive distance data. *Bull. Math. Biol.* **51**, 597–603 (1989)
10. King, V., Zhang, L., Zhou, Y.: On the complexity of distance-based evolutionary tree construction. In: *Proceedings of the 14<sup>th</sup> Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2003)*, pp. 444–453 (2003)
11. Nakhleh, L., Warnow, T., Ringe, D., Evans, S.N.: A comparison of phylogenetic reconstruction methods on an Indo-European dataset. *Trans. Philol. Soc.* **103**, 171–192 (2005)
12. Reyzin, L., Srivastava, N.: On the longest path algorithm for reconstructing trees from distance matrices. *Inf. Process. Lett.* **101**, 98–100 (2007)
13. The Canterbury Tales Project: University of Birmingham, Brigham Young University, University of Münster, New York University, Virginia Tech, and Keio University. Website: <http://www.canterburytalesproject.org/>
14. Warnow, T.: Some combinatorial optimization problems in phylogenetics. In: Lovász, L., Gyárfás, G., Katona, G., Recski, A., Székely, L. (eds.) *Graph Theory and Combinatorial Biology*. Bolyai Society Mathematical Studies, vol. 7, pp. 363–413. Bolyai János Matematikai Társulat (1999)
15. Waterman, M.S., Smith, T.F., Singh, M., Beyer, W.A.: Additive evolutionary trees. *J. Theor. Biol.* **64**, 199–213 (1977)
16. Wu, B.Y., K.-Chao, M., Tang, C.Y.: Approximation and exact algorithms for constructing minimum ultrametric trees from distance matrices. *J. Combin. Optim.* **3**, 199–211 (1999)

## Phylogeny Reconstruction

- ▶ Distance-Based Phylogeny Reconstruction (Optimal Radius)

## Planar Geometric Spanners

2005; Bose, Smid, Gudmundsson

JOACHIM GUDMUNDSSON<sup>1</sup>, GIRI NARASIMHAN<sup>2</sup>,  
MICHIEL SMID<sup>3</sup>

<sup>1</sup> DMiST, National ICT Australia Ltd, Alexandria, Australia

<sup>2</sup> Department of Computer Science, Florida International University, Miami, FL, USA

<sup>3</sup> School of Computer Science, Carleton University, Ottawa, ON, Canada