26. Lavi R, Swamy C (2005) Truthful and near-optimal mechanism design via linear programming. In: Proceedings of the 46th annual symposium on foundations of computer science (FOCS), pp 595–604
27. Lavi R, Swamy C (2007) Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity. Working paper
28. Lehmann B, Lehmann D, Nisan N (2006) Combinatorial auctions with decreasing marginal utilities. Games Econ Behav 55(2):270–296
29. Lehmann D, O'Callaghan L, Shoham Y (2002) Truth revelation in approximately efficient combinatorial auctions. J ACM 49(5):577–602
30. Mu'alem A, Schapira M (2007) Setting lower bounds on truthfulness. In: Proceedings of the 18th symposium on discrete algorithms (SODA)
31. Nisan N, Ronen A (2000) Computationally feasible VCG mechanisms. In: Proceedings of the 2nd ACM conference on electronic commerce (EC'00)
32. Nisan N, Ronen A (2001) Algorithmic mechanism design. Games Econ Behav 35:166–196
33. Nisan N, Roughgarden T, Tardos E, Vazirani V (2007) Algorithmic game theory. Cambridge University Press (expected to appear)
34. Roberts K (1979) The characterization of implementable choice rules. In: Laffont JJ (ed) Aggregation and revelation of preferences. North-Holland, pp 321–349
35. Saks M, Yu L (2005) Weak monotonicity suffices for truthfulness on convex domains. In: Proceedings of the 6th ACM conference on electronic commerce (ACM-EC), pp 286–293

Algorithms for Combining Rooted Triplets into a Galled Phylogenetic Network

Jesper Jansson[1] and Wing-Kin Sung[2]
[1]Laboratory of Mathematical Bioinformatics, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto, Japan
[2]Department of Computer Science, National University of Singapore, Singapore, Singapore

Keywords

Dense set; Galled phylogenetic network; Phylogenetic tree; Polynomial-time approximation algorithm; Rooted triplet

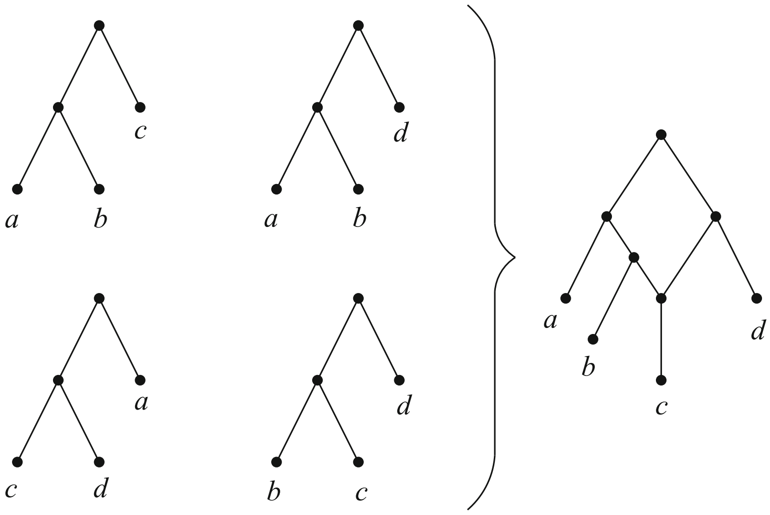
Years and Authors of Summarized Original Work

2006; Jansson, Sung
2006; Jansson, Nguyen, Sung
2006; He, Huynh, Jansson, Sung
2010; Byrka, Gawrychowski, Huber, Kelk
2011; van Iersel, Kelk

Problem Definition

A *phylogenetic tree* is a binary, rooted, unordered tree whose leaves are distinctly labeled. A *phylogenetic network* is a generalization of a phylogenetic tree formally defined as a rooted, connected, directed acyclic graph in which (1) each node has outdegree at most 2; (2) each node has indegree 1 or 2, except the root node which has indegree 0; (3) no node has both indegree 1 and outdegree 1; and (4) all nodes with outdegree 0 are labeled by elements from a finite set L in such a way that no two nodes are assigned the same label. Nodes of outdegree 0 are referred to as *leaves* and are identified with their corresponding elements in L. Nodes with indegree 2 are called *reticulation nodes*. For any phylogenetic network N, let $\mathcal{U}(N)$ be the undirected graph obtained from N by replacing each directed edge by an undirected edge. N is said to be a *galled phylogenetic network* (*galled network*, for short) if all cycles in $\mathcal{U}(N)$ are node-disjoint. Galled networks are also known in the literature as *topologies with independent recombination events* [15], *galled-trees* [6], and *level*-1 *phylogenetic networks* [2, 5, 7, 9, 10, 14].

A phylogenetic tree with exactly three leaves is called a *rooted triplet*. The unique rooted triplet on a leaf set $\{x, y, z\}$ in which the lowest common ancestor of x and y is a proper descendant of the lowest common ancestor of x and z (or equivalently, where the lowest common ancestor of x and y is a proper descendant of the lowest common ancestor of y and z) is denoted by $xy|z$. For any phylogenetic network N, the rooted triplet $xy|z$ is said to be *consistent* with N if N contains three leaves labeled by x, y, and z



Algorithms for Combining Rooted Triplets into a Galled Phylogenetic Network, Fig. 1 A dense set $\mathcal{T} = \{ab|c,\ ab|d,\ cd|a,\ bc|d\}$ of rooted triplets with leaf set $\{a, b, c, d\}$ and a galled phylogenetic network that is consistent with $\mathcal{T}$. Note that this solution is not unique

as well as two internal vertices w and z such that there are four directed paths of nonzero length from w to a, from w to b, from z to w, and from z to c that are vertex-disjoint except for in the vertices w and z. A set $\mathcal{T}$ of rooted triplets is *consistent* with N if every rooted triplet in $\mathcal{T}$ is consistent with N. See Fig. 1 for an example.

Denote the set of leaves in any phylogenetic network N by $\Lambda(N)$, and for any set $\mathcal{T}$ of rooted triplets, define $\Lambda(\mathcal{T}) = \bigcup_{t_i \in \mathcal{T}} \Lambda(t_i)$. A set $\mathcal{T}$ of rooted triplets is *dense* if for each $\{x, y, z\} \subseteq \Lambda(\mathcal{T})$, at least one of the three possible rooted triplets $xy|z$, $xz|y$, and $yz|x$ belongs to $\mathcal{T}$. Observe that if $\mathcal{T}$ is dense, then $|\mathcal{T}| = \Theta(|\Lambda(\mathcal{T})|^3)$. Jansson and Sung introduced the following problem in [10].

Problem 1 Given a set $\mathcal{T}$ of rooted triplets, output a galled network N with $\Lambda(N) = \Lambda(\mathcal{T})$ such that N and $\mathcal{T}$ are consistent, if such a network exists; otherwise, output *null*.

A natural optimization version of Problem 1 is:

Problem 2 Given a set $\mathcal{T}$ of rooted triplets, output a galled network N with $\Lambda(N) = \Lambda(\mathcal{T})$ that is consistent with the maximum possible number of rooted triplets belonging to $\mathcal{T}$.

A generalization of Problem 1 studied by He et al. in [8] involves *forbidden* rooted triplets and is defined as follows.

Problem 3 Given two sets $\mathcal{T}$ and $\mathcal{F}$ of rooted triplets, output a galled network N with $\Lambda(N) = \Lambda(\mathcal{T}) \cup \Lambda(\mathcal{F})$ such that (1) N and $\mathcal{T}$ are consistent and (2) N is not consistent with any rooted triplet belonging to $\mathcal{F}$; if no such network exists, output *null*.

Below, we write $L = \Lambda(\mathcal{T})$ and $n = |L|$.

Key Results

As shown in [11], Problem 1 can be solved in (optimal) $O(|\mathcal{T}|) = O(n^3)$ time for dense inputs:

Theorem 1 ([11]) *Given any dense set $\mathcal{T}$ of rooted triplets with leaf set L, a galled network consistent with $\mathcal{T}$ (if one exists) can be constructed in $O(n^3)$ time, where $n = |L|$.*

The algorithm referred to in Theorem 1 was extended by van Iersel and Kelk [14] as follows.

Theorem 2 ([14]) *Given any dense set $\mathcal{T}$ of rooted triplets with leaf set L, a galled network consistent with $\mathcal{T}$ (if one exists) that contains as few reticulation nodes as possible can be constructed in $O(n^5)$ time, where $n = |L|$.*

For the more general case of nondense inputs, Problem 1 becomes harder:

Theorem 3 ([11]) *The problem of determining if there exists a galled network that is consistent* *with an input nondense set $\mathcal{T}$ of rooted triplets is NP-hard.*

Since not all sets of rooted triplets are consistent with a galled network, it is of interest to consider Problem 2. It follows from Theorem 3 that Problem 2 is also NP-hard for nondense inputs, and this motivates polynomial-time approximation algorithms. Say that an algorithm for Problem 2 is an *f-approximation algorithm* if it always returns a galled network N such that $\frac{N(\mathcal{T})}{|\mathcal{T}|} \geq f$, where $N(\mathcal{T})$ is the number of rooted triplets in $\mathcal{T}$ that are consistent with N. Define the nonlinear recurrence relation $S(n) = \max_{1 \leq a \leq n}\left\{\binom{a}{3} + 2\cdot\binom{a}{2}\cdot(n-a) + a\cdot\binom{n-a}{2} + S(n-a)\right\}$ for $n > 0$ and $S(0) = 0$. It was shown in [4] that $\lim_{n\to\infty} \frac{S(n)}{3\binom{n}{3}} = \frac{2(\sqrt{3}-1)}{3} \approx 0.488033\ldots$ and that $\frac{S(n)}{3\binom{n}{3}} > \frac{2(\sqrt{3}-1)}{3} \approx 0.488033\ldots$ for all $n > 2$. The following theorem was proved by Byrka et al. in [2].

Theorem 4 ([2]) *There exists an $\frac{S(n)}{3\binom{n}{3}}$-approximation algorithm for Problem 2 that runs in $O(n^3 + n|\mathcal{T}|)$ time.*

A matching negative bound is:

Theorem 5 ([11]) *For any $f > \lim_{n\to\infty} \frac{S(n)}{3\binom{n}{3}}$, there exists a set $\mathcal{T}$ of rooted triplets such that no galled network can be consistent with at least a factor of f of the rooted triplets in $\mathcal{T}$. (Thus, no f-approximation algorithm for Problem 2 is possible.)*

For Problem 3, Theorem 3 immediately implies NP-hardness by taking $\mathcal{F} = \emptyset$. The following positive result is known for the optimization version of Problem 3.

Theorem 6 ([8]) *There exists an $O(|L|^2|\mathcal{T}|(|\mathcal{T}| + |\mathcal{F}|))$-time algorithm for inferring a galled network N that guarantees $|N(\mathcal{T})| - |N(\mathcal{F})| \geq \frac{5}{12}\cdot(|\mathcal{T}| - |\mathcal{F}|)$, where $L = \Lambda(\mathcal{T}) \cup \Lambda(\mathcal{F})$.*

Finally, we remark that the analogous version of Problem 1 of inferring a phylogenetic *tree* consistent with all the rooted triplets in an input set (when such a tree exists) can be solved in polynomial time with a classical algorithm by Aho et al. [1] from 1981. Similarly, for Problem 2, to infer a phylogenetic tree consistent with as many rooted triplets from an input set of rooted triplets as possible is NP-hard and admits a polynomial-time $1/3$-approximation algorithm, which is optimal in the sense that there exist certain inputs for which no tree can achieve a factor larger than $1/3$. See, e.g., [3] for a survey of known results about maximizing rooted triplet consistency for trees. On the other hand, more complex network structures such as the *level-k phylogenetic networks* [5] permit a higher percentage of the input rooted triplets to be embedded; in the extreme case, if there are no restrictions on the reticulation nodes at all, then a sorting network-based construction yields a phylogenetic network that is trivially consistent with every rooted triplet over L [10]. A number of efficient algorithms for combining rooted triplets into higher level networks have been developed; see, e.g., [2,7,14] for further details and references.

Applications

Phylogenetic networks are used by scientists to describe evolutionary relationships that do not fit the traditional models in which evolution is assumed to be treelike. Evolutionary events such as horizontal gene transfer or hybrid speciation (often referred to as *recombination events*) which suggest convergence between objects cannot be represented in a single tree but can be modeled in a phylogenetic network as internal nodes having more than one parent (i.e., reticulation nodes). The phylogenetic network is a relatively new tool, and various fast and reliable methods for constructing and comparing phylogenetic networks are currently being developed.

Galled networks form an important class of phylogenetic networks. They have attracted special attention in the literature [5, 6, 15] due to their biological significance (see [6]) and their simple, almost treelike, structure. When the number of recombination events is limited and most of the recombination events have occurred recently, a galled network may suffice to accurately describe the evolutionary process under study [6]. The motivation behind the rooted triplet approach taken here is that a highly accurate tree for each cardinality-three subset of the leaf set can be obtained through maximum likelihood-based methods or Sibley-Ahlquist-style DNA-DNA hybridization experiments (see [13]). The algorithms mentioned above can be used as the merging step in a divide-and-conquer approach for constructing phylogenetic networks analogous to the quartet method paradigm for inferring unrooted phylogenetic trees [12] and other supertree methods. We consider dense input sets in particular because this case can be solved in polynomial time.

Open Problems

The approximation factor given in Theorem 4 is expressed in terms of the number of rooted triplets in the input $\mathcal{T}$, and Theorem 5 shows that it cannot be improved. However, if one measures the quality of the approximation in terms of a galled network N_{OPT} that is consistent with the maximum possible number of rooted triplets from $\mathcal{T}$, Theorem 4 can be far from optimal. An open problem is to determine the polynomial-time approximability and inapproximability of Problem 2 when the approximation ratio is defined as $\frac{N(\mathcal{T})}{N_{OPT}(\mathcal{T})}$ instead of $\frac{N(\mathcal{T})}{|\mathcal{T}|}$.

Another research direction is to develop fixed-parameter polynomial-time algorithms for Problem 1. The level of the constructed network, the number of allowed reticulation nodes, or some measure of the density of the input set of rooted triplet might be suitable parameters.

URLs to Code and Data Sets

A Java implementation of the algorithm for Problem 1 referred to in Theorem 2 (coded by its authors [14]) is available at http://skelk.sdf-eu.org/marlon.html. See also http://skelk.sdf-eu.org/lev1athan/ for a Java implementation of a polynomial-time heuristic described in [9] for Problem 2.

Cross-References

- ▸ Directed Perfect Phylogeny (Binary Characters)
- ▸ Distance-Based Phylogeny Reconstruction (Fast-Converging)
- ▸ Distance-Based Phylogeny Reconstruction: Safety and Edge Radius
- ▸ Perfect Phylogeny (Bounded Number of States)
- ▸ Phylogenetic Tree Construction from a Distance Matrix

Acknowledgments JJ was funded by the Hakubi Project at Kyoto University and KAKENHI grant number 26330014.

Recommended Reading

1. Aho AV, Sagiv Y, Szymanski TG, Ullman JD (1981) Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. SIAM J Comput 10(3):405–421
2. Byrka J, Gawrychowski P, Huber KT, Kelk S (2010) Worst-case optimal approximation algorithms for maximizing triplet consistency within phylogenetic networks. J Discret Algorithms 8(1):65–75
3. Byrka J, Guillemot S, Jansson J (2010) New results on optimizing rooted triplets consistency. Discret Appl Math 158(11):1136–1147
4. Chao K-M, Chu A-C, Jansson J, Lemence RS, Mancheron A (2012) Asymptotic limits of a new type of maximization recurrence with an application to bioinformatics. In: Proceedings of the 9th annual conference on theory and applications of models of computation (TAMC 2012), Beijing, pp 177–188
5. Choy C, Jansson J, Sadakane K, Sung W-K (2005) Computing the maximum agreement of phylogenetic networks. Theor Comput Sci 335(1):93–107
6. Gusfield D, Eddhu S, Langley C (2003) Efficient reconstruction of phylogenetic networks with constrained recombination. In: Proceedings of computational systems bioinformatics (CSB2003), Stanford, pp 363–374
7. Habib M, To T-H (2012) Constructing a minimum phylogenetic network from a dense triplet set. J Bioinform Comput Biol 10(5):1250013
8. He Y-J, Huynh TND, Jansson J, Sung W-K (2006) Inferring phylogenetic relationships avoiding forbidden rooted triplets. J Bioinform Comput Biol 4(1):59–74
9. Huber KT, van Iersel L, Kelk S, Suchecki R (2011) A practical algorithm for reconstructing level-1 phylogenetic networks. IEEE/ACM Trans Comput Biol Bioinform 8(3):635–649

10. Jansson J, Sung W-K (2006) Inferring a level-1 phylogenetic network from a dense set of rooted triplets. Theor Comput Sci 363(1):60–68
11. Jansson J, Nguyen NB, Sung W-K (2006) Algorithms for combining rooted triplets into a galled phylogenetic network. SIAM J Comput 35(5):1098–1121
12. Jiang T, Kearney P, Li M (2001) A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application. SIAM J Comput 30(6):1942–1961
13. Kannan S, Lawler E, Warnow T (1996) Determining the evolutionary tree using experiments. J Algorithms 21(1):26–50
14. van Iersel L, Kelk S (2011) Constructing the simplest possible phylogenetic network from triplets. Algorithmica 60(2):207–235
15. Wang L, Zhang K, Zhang L (2001) Perfect phylogenetic networks with recombination. J Comput Biol 8(1):69–78

All Pairs Shortest Paths in Sparse Graphs

Seth Pettie
Electrical Engineering and Computer Science (EECS) Department, University of Michigan, Ann Arbor, MI, USA

Keywords

Quickest route; Shortest route

Years and Authors of Summarized Original Work

2004; Pettie

Problem Definition

Given a communications network or road network, one of the most natural algorithmic questions is how to determine the shortest path from one point to another. The *all pairs* shortest path problem (APSP) is, given a directed graph $G = (V, E, l)$, to determine the distance and shortest path between every pair of vertices, where $|V| = n, |E| = m$, and $l : E \to \mathbb{R}$ is the edge length (or weight) function. The output is in the form of two $n \times n$ matrices: $D(u, v)$ is the distance from u to v and $S(u, v) = w$ if (u, w) is the first edge on a shortest path from u to v. The APSP problem is often contrasted with the *point-to-point* and *single source* (SSSP) shortest path problems. They ask for, respectively, the shortest path from a given source vertex to a given target vertex and all shortest paths from a given source vertex.

Definition of Distance

If ℓ assigns only non-negative edge lengths then the definition of distance is clear: $D(u, v)$ is the length of the minimum length path from u to v, where the length of a path is the total length of its constituent edges. However, if ℓ can assign negative lengths then there are several sensible notations of distance that depend on how negative length cycles are handled. Suppose that a cycle C has negative length and that $u, v \in V$ are such that C is reachable from u and v reachable from C. Because C can be traversed an arbitrary number of times when traveling from u to v, there is no shortest path from u to v using a finite number of edges. It is sometimes assumed a priori that G has no negative length cycles; however it is cleaner to define $D(u, v) = -\infty$ if there is no finite shortest path. If $D(u, v)$ is defined to be the length of the shortest *simple* path (no repetition of vertices) then the problem becomes NP-hard. (If all edges have length -1 then $D(u, v) = -(n-1)$ if and only if G contains a Hamiltonian path [7] from u to v.) One could also define distance to be the length of the shortest path without repetition of edges.

Classic Algorithms

The Bellman-Ford algorithm solves SSSP in $O(mn)$ time and under the assumption that edge lengths are non-negative, Dijkstra's algorithm solves it in $O(m + n \log n)$ time. There is a well known $O(mn)$-time shortest path preserving transformation that replaces any length function with a non-negative length function. Using this transformation and n runs of Dijkstra's algorithm gives an APSP algorithm running in $O(mn + n^2 \log n) = O(n^3)$ time. The Floyd-Warshall algorithm computes APSP in a more