Supertree Construction with Constraint Programming

Ian P. Gent[1], Patrick Prosser[2], Barbara M. Smith[3], and Wu Wei[2]

[1] School of Computer Science, University of St. Andrews, Scotland.
ipg@dcs.st-and.ac.uk
[2] Department of Computing Science, University of Glasgow, Scotland.
pat@dcs.gla.ac.uk
[3] School of Computing and Engineering, University of Huddersfield, England.
b.m.smith@hud.ac.uk

1 Introduction

A central goal of systematics is the construction of a *tree of life*, where the tree represents the relationship between all living things. The leaf nodes of the tree correspond to species and the internal nodes to hypothesized species, assumed to be extinct, where species have diverged. One problem that biologists face is to assemble a *supertree* from many smaller trees that have overlapping leaf sets. Polytime algorithms have been proposed for this problem [3, 5]. We present a simple constraint encoding of this problem. This is based on the observation that any rooted tree can be considered as being min-ultrametric when we label interior nodes with their depth in that tree. That is, any path from the root to a leaf corresponds to a strictly increasing sequence. Our encoding takes a radically different approach to solving these problems, and represents a new perspective.

2 Species Trees and Supertrees

In a fully resolved (i.e. bifurcating) species tree each internal node has degree 3, with the exception of the root. Consequently a fully resolved species tree with n leaf nodes has $n-1$ internal nodes. In Figure 1 we have three species, namely a, b, and c. Species a and b are more closely related to one another than they are to c. More specifically, we say that *the most recent common ancestor of a and b is greater than the most recent common ancestor of a and c (equally b and c)*, where the most recent common ancestor of two leaf nodes a and b is the internal node furthest from the root that has a and b as descendants. We compare most recent common ancestors by measuring their distance from the root. That is,

$$mrca(a,b) > mrca(a,c) \tag{1}$$

$$mrca(a,b) > mrca(b,c) \tag{2}$$

$$mrca(a,c) = mrca(b,c) \tag{3}$$

Note, that in Figure 1 we have labeled the two interior nodes. Generally, interior nodes are anonymous and we introduce this labeling only to explain the equations above. We see that the most recent common ancestor of a and b is interior node Y, i.e. $mrca(a, b) = Y$. Furthermore, $mrca(a, b) = mrca(b, a) = Y$ (i.e. the relation is symmetric) and from equation (3) $mrca(a, c) = mrca(b, c) = X$. From equation (1) we have $mrca(a, b) > mrca(a, c)$ (i.e. $Y > X$) and $mrca(a, b) > mrca(b, c)$ (i.e. yet again $Y > X$ from equation (2)).

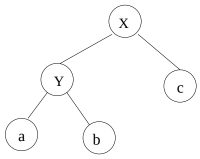


Fig. 1. A species tree, where species a and b are more closely related to each other than they are to species c. This small tree can also be represented as the rooted triple $((a, b), c)$.

Species trees are frequently presented as a collection of rooted triples, of the form $((a, b), c)$ meaning that $mrca(a, b) > mrca(a, c)$, $mrca(a, b) > mrca(b, c)$, and $mrca(a, c) = mrca(b, c)$. The *BreakUp* algorithm of [3] takes as input a species tree and delivers as a result a set of rooted triples that define that tree. The $OneTree$ algorithm [3, 1] takes these triples as input and produces a supertree, if one exists. In Figure 2 two trees are combined to produce a supertree. First, the two initial trees are broken up into rooted triples using the *BreakUp* algorithm. These triples are then passed to $OneTree$ along with the set of species $\{a, b, c, d, e, f, g\}$, and a supertree (f,((d,e),((c,(a,b)),g))) is produced. Note that there are 9 possible supertrees that respect those triples [6]. The $OneTree$ algorithm is of complexity $O(m.n)$ where we have n leaf nodes and m triples.

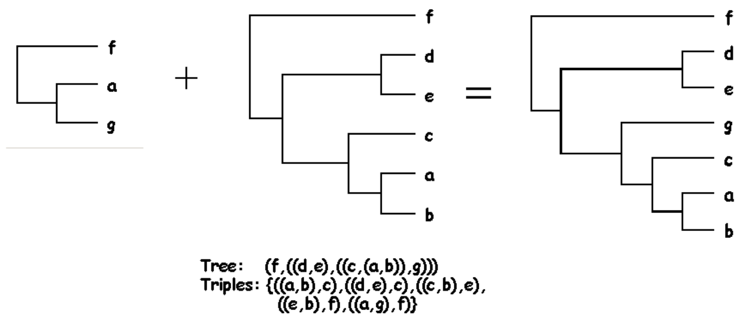


Fig. 2. $OneTree$ applied to the two trees (f,(a,g)) and (f,((d,e),(c,(a,b)))), giving the supertree (f,((d,e),((c,(a,b)),g))). $OneTree$ takes as input the set of rooted triples $R = \{((a, g), f), ((a, b), c), ((d, e), c), ((c, b), e), ((e, b), f)\}$ and the set of species $S = \{a, b, c, d, e, f, g\}$

Definition 1. *An* **ultrametric tree** T *is a rooted tree with* n *uniquely labeled leaves and interior nodes labeled with values (real or integer) such that any path from the root to a leaf node constitutes a strictly* decreasing *sequence. In a* **min-ultrametric tree** *interior nodes are labeled with values such that any path from root to a leaf is a strictly* increasing *sequence [2].*

An ultrametric tree can be represented by an ultrametric matrix D, and an ultrametric matrix can be represented by an ultrametric tree.

Definition 2. *Let* D *be an* $n \times n$ *symmetric matrix.* D *is an* **ultrametric matrix** *if there are at most* $n-1$ *distinct values within* D*, and for any three indices* i*,* j*, and* k *there is a tie for the maximum of* $D_{i,j}$*,* $D_{i,k}$*, and* $D_{j,k}$*.*

In Figure 3 we have an ultrametric matrix D and the corresponding ultrametric tree T (these figures are taken from [2] page 450). Looking at the tree and matrix we see that the most recent common ancestor of A and D is labeled with the value 5, and the most recent common ancestor of E and B is labeled with value 8. The leaf nodes might be considered as species, and the values on the interior nodes the number (let's say) of millions of years that have passed since species diverged (i.e. species D diverged from species E 5 million years ago). An ultrametric tree can be constructed from an ultrametric matrix in time $O(n^2)$ [2].

	A	B	C	D	E
A	0	8	8	5	3
B		0	3	8	8
C			0	8	8
D				0	5
E					0

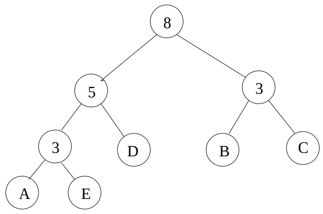


Fig. 3. The ultrametric matrix D and its ultrametric tree T

2.1 The Min-Ultrametric Constraint

We now present a constraint encoding which provides a unique representation of trees up to symmetry of either renaming of internal nodes, or swapping of left and right subtrees at any node. We encode the *depth of* the most recent common ancestors in the tree. We have a $n \times n$ two dimensional array D of constrained integer variables. Each variable $D_{i,j}$ takes a value in the range 0 to $n-1$. Since the array is symmetric, $D_{i,j} = D_{j,i}$, for all i and j, and we arbitrarily set the diagonal $D_{i,i} = 0$. The value assigned to $D_{i,j}$ represents the depth of the most recent common ancestor of leaf nodes i and j. To encode the constraints on D, we first associate the integers 1 to n with the species and start with the constraint for the triple $((a,b),c)$.

$$triple(a,b,c) \equiv [(D_{a,c} = D_{b,c}) \wedge (D_{a,b} > D_{b,c}) \wedge (D_{a,b} > D_{a,c})] \tag{4}$$

This encodes the equations (1), (2), and (3). We can then guarantee that D is a min-ultrametric matrix by demanding that:

$$\forall a \in \{1..n-2\}.\ \forall b \in \{a+1..n-1\}.\ \forall c \in \{b+1..n\} \\ (triple(a,b,c) \vee triple(b,c,a) \vee triple(c,a,b)) \qquad (5)$$

i.e. one of these triples holds for every combination of three species a, b, and c. This successfully encodes that the minimum value of the three variables $D_{a,b}$, $D_{b,c}$, $D_{c,a}$ is shared by two of them and not the third, the defining property of a min-ultrametric matrix where the resultant tree is bifurcating. This can also be viewed from a geometric perspective. We can consider the indices a, b and c as being vertices of a triangle, and the matrix elements $D_{a,b}$, $D_{a,c}$ and $D_{b,c}$ as being the length of the edges of that triangle. Triangles are forced to be isosceles, and equilateral triangles are disallowed.

We need to insist that the values in a row of D do not contain any gaps. That is, in the resultant tree any path from the root to a leaf node will be an increasing sequence, and that sequence will have no numeric gaps. We do this by introducing arrays of constrained integer variables to count the number of occurrences of each integer in each row of D. We then demand that if the count for i is zero, then the count for $i+1$ is also zero. Equivalently, if i occurs in a row, and i is greater than 0, then $i-1$ occurs in that row also, i.e.

$$\forall\ a\ \forall\ b\ [(D_{a,b} = i\ \wedge\ i > 0) \rightarrow \exists\ c\ (D_{a,c} = i-1)] \qquad (6)$$

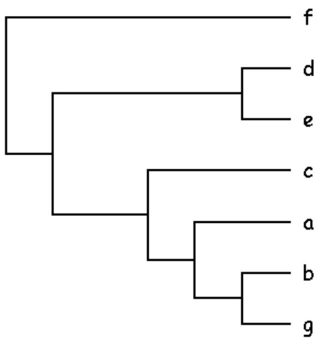
With this encoding, any consistent instantiation of the variables in D is min-ultrametric and has a min-ultrametric tree. The number of possible such instantiations is $\frac{(2n-2)!}{2^{n-1}(n-1)!}$ [4].

Given a set of rooted triples R we can then post each triple as a constraint on the array D. For a triple $((i,j),k) \in R$ we post the constraint $triple(i,j,k)$ from equation (4) above. This breaks the three disjunctive constraints already posted between these three variables in (5) above. Consequently, a consistent instantiation of the variables in D will correspond to a species tree that respects the triples in R. Given that consistent instantiation of D we can then process D to construct the tree, using the algorithm in [2]. Therefore, our constraint encoding achieves the same net result as the $OneTree$ algorithm, and trivially extends to the enumeration of all species trees (i.e. performing the same function as $AllTrees$ [3]). To generate all trees we allow our solving procedure to backtrack whenever it finds a solution, and to continue on to the next solution.

Our encoding only produces bifurcating trees, whereas $OneTree$ may produce trees with interior nodes with more than two children. This happens when the triples do not fully define the tree, whereas the constraint encoding automatically forces a resolution. We can relax our constraints to allow this. The disjunctive constraints posted across the matrix D can be relaxed as follows:

$$relTriple(a,b,c) \equiv [(D_{a,c} = D_{b,c})\ \wedge\ (D_{a,b} \geq D_{b,c})\ \wedge\ (D_{a,b} \geq D_{a,c})] \qquad (7)$$

In equation (5) replace $triple(a,b,c)$ with our relaxed constraint $relTriple(a,b,c)$. From a geometric perspective, for any three vertices a, b, c, by default we allow



	a	b	c	d	e	f	g
a	0	4	3	2	2	1	4
b	4	0	3	2	2	1	5
c	3	3	0	2	2	1	3
d	2	2	2	0	3	1	2
e	2	2	2	3	0	1	2
f	1	1	1	1	1	0	1
g	4	5	3	2	2	1	0

Fig. 4. The min-ultrametric matrix and tree produced from the rooted triples $\{((a,g),f),((a,b),c),((d,e),c),((c,b),e),((e,b),f)\}$. Note that there are again 9 possible solutions [6]. This tree is comparable to that given in Figure 2

triangles to be isosceles or equilateral. Our constraint encoding requires $O(n^2)$ variables, each with a domain of size n, and $O(n^3)$ ternary constraints. Figure 4 shows the species tree with its ultrametric matrix, resulting from the constraint encoding of the trees in Figure 2.

3 Conclusion and Future Work

We have presented a new method for building supertrees. This is based on the observations that any tree can be considered min-ultrametric when we label interior nodes with their depth in that tree, and any min-ultrametric matrix has a corresponding min-ultrametric tree. This then leads us to an encoding of the decision problem as a constraint satisfaction problem. This can be extended to deal with unresolved triples as well as over-constrained problems. We also expect that our encoding can be made more efficient.

References

1. D. Bryant and M. Steel. Extension Operations on Sets of Leaf-labeled Trees. *Advances in Applied Mathematics*, 16:425–453, 1995.
2. Dan Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, 1997.
3. Meei Pyng Ng and Nicholas C. Wormald. Reconstruction of rooted trees from subtrees. *Discrete Applied Mathematics*, 69:19–31, 1996.
4. E. Schröder. Vier combinatorische probleme. *Zeit. für Math. Phys.*, 15:361–376, 1870.
5. Charles Semple and Mike Steel. A supertree method for rooted trees. *Discrete Applied Mathematics*, 105:147–158, 2000.
6. Joseph L. Thorley. Cladistic information, leaf stability and supertree construction. *PhD Thesis, University of Bristol, Department of Biological Sciences*, 2000.