Application of Ant Colony Optimization for Mapping the Combinatorial Phylogenetic Search Space

Alexander Safatli[1] and Christian Blouin[1,2]

[1]*Faculty of Computer Science, Dalhousie University, Halifax, Canada*

[2]*Department of Biochemistry & Molecular Biology, Dalhousie University, Halifax, Canada*

Keywords: Phylogenetic Trees, Combinatorial Space, Ant Colony Optimization, Swarm Intelligence.

Abstract: In bioinformatics, landscapes of phylogenetic trees for an alignment of sequence data are defined by a discrete state combinatorial space. The optimal solution in such a space is the best-fitting tree which provides insight on the evolutionary relationship between taxonomic groups. The underlying structure of this space is poorly understood. The Ant Colony Optimization (ACO) algorithm is applied in a novel manner to sample phylogenetic tree landscapes in order to understand more about this structure. The proposed implementation provides a probabilistic model for exploring this combinatorial space. This probabilistic model allows us to circumvent the complexity that arises due to increasing the number of sequences. In order to evaluate its performance, quantities of resultant solutions were judged in order to determine how much of the space can be sampled. The results show that the algorithm is robust to the starting location and consistently samples a majority of the search space.

1 INTRODUCTION

Trees have been used to a great extent in biology to graphically represent evolutionary relationships between species, subpopulations, and genes. These trees are generally semi-labelled, binary, and rooted, but they can also be multifurcating or unrooted. They descend from a root node, bifurcate at other nodes and end at leaves labelled by the names of operational taxonomic units (OTUs). Internal nodes are thus inferred ancestors of these OTUs. Traditionally, these trees were inferred from morphological similarities, where two species that shared the most characteristics were considered siblings or homologous and shared a common ancestor that was not the ancestor of any other species. More recently, the inference of these so-called phylogenetic trees is done from sequence alignment data.

This inference or tree reconstruction is a central problem in computational biology and is known to be NP-complete (Foulds and Graham, 1982). Biologists have to therefore use approximate optimization algorithms with certain starting points and certain random moves between trees where resulting trees can vary from run to run. Many such methods exist for the construction of phylogenetic trees. All share a fitness function f that scores the fit of a particular tree with a sequence alignment. Examples of fitness functions include likelihood and parsimony (Fitch, 1971).

A phylogenetic landscape or tree space refers to the combinatorial space of all possible phylogenetic tree topologies for a set of n leaves or taxa. This forms a discrete solution search space and finite graph $G = (T, E)$ viewed as a function of its vertices. The finite set E refers to the neighborhood relation on T, a set of tree topologies or configurations (Bastert et al., 2002; Charleston, 1995; Stadler, 1996). Transitions between states in this space are bidirectional and represent transformation from one tree topology to another by a tree rearrangement operator. We use the the Subtree Prune and Regraft (SPR) operator which describes a transfer of a node in a phylogenetic tree from one parent node to a new parent (Felsenstein, 2004). The phylogenetic tree space is central to an example of a combinatorial optimization problem $P = (T, f)$. The set of all possible feasible assignments T is known as the search space. Each element of the set is a candidate solution. The optimal solution in tree space is the solution $t^* \in T$ with maximum fitness function value $f(t^*) \geq f(t) \forall t \in T$.

Because the structure of this space is still poorly understood, and distinguishing elements of this space are still being discovered (Sanderson et al., 2011), we sought to find a method to sample large areas of

Safatli A. and Blouin C..
Application of Ant Colony Optimization for Mapping the Combinatorial Phylogenetic Search Space.
DOI: 10.5220/0005255701950200
In *Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms* (BIOINFORMATICS-2015), pages 195-200
ISBN: 978-989-758-070-3

this space in an effort to discover large numbers of trees that comprise them, the structure of the graph that connects these trees, and how this configuration of trees and edges can be utilized to improve existing approximate methods. To this end, we chose the Ant Colony Optimization (ACO) algorithm and implemented for use in exploring the space so as to sample it for trees and structure.

The ACO algorithm is a metaheuristic related to swarm intelligence. It models the foraging behaviour of ants with a collective behaviour to find paths between food sources and their nests. The biological mechanism for this involves the deposition of a certain concentration of pheromones by ants as they search for food based on whether or not a path they are on leads to food. In the ACO algorithm, edges in a graph are paths, and paths of higher pheromone concentrations are chosen with higher probability by ant agents as they traverse the space. Therefore, cooperative interaction emerges to solve for shortest paths through a parametrized probabilistic model (Blum and Roli, 2003). A solution to the optimization problem can be expressed in terms of a feasible path, or ant trail, on this graph (Blum and Roli, 2003; Dorigo et al., 1999; Luke, 2013).

The first phase of the algorithm involves ant generation and activity. This includes the stochastic movement of ants based on edge weights. The second phase is when pheromone evaporation takes place (Dorigo et al., 1999). Pheromone evaporation describes the decay of pheromone intensity over time. This avoids too rapid convergence towards a suboptimal region (Dorigo et al., 1999).

The characteristic of a candidate solution is left problem-specific (Luke, 2013). While moving, ants keep in memory the path it follows. This forms a partial solution to the problem. When an ant has built a candidate solution to the problem, it will retrace its path back to the source node and die (Dorigo et al., 1999).

2 METHODS

2.1 PLACO

Our primary objective is to sample the phylogenetic tree space. Therefore, sampling of the space should avoid poorly scoring regions unless such regions are necessary for ants to visit in order to move to a region of near globally optimal fitness. In order to do so in an effective and timely manner, it would also be beneficial to avoid consideration of all possible transformations for a topology and only sample a subset of them.

We construct an algorithm based on the simple ant colony optimization metaheuristic introduced in (Dorigo et al., 1999). In this algorithm, labelled as the Phylogenetic Landscape Ant Colony Optimization (PLACO) algorithm, a single population of ant agents explore a space G by visiting its trees. Pheromone trails are expressed as edge weights proportional to the change in f between one tree and the next. Each ant in the system performs a random walk to adjacent vertices in the graph until they run out of a quantity we define as energy. An ant that runs out of energy returns to the colony by retracing its path and subsequently dies.

In both directions that an ant travels, away from and back to the colony, pheromone trails are deposited as an increase to edge weights. This corresponds with topology fitness along the forward path, but along the path in the return trip this is a function of the fittest topology visited. This can be computed to be equal to an exponentiation of the change in $f(t)$ from an origin topology to the next. We present the weight on the edge between t_i and t_j as $\Delta w_{ij} = 2^{(f(t_j)-f(t_i))}$. Negative changes in the fitness function will result in a smaller concentration of pheromones to be deposited by ants.

Avoiding the computation of all neighboring reconfigurations to a topology can allow this heuristic to effectively scale to large n, as the degree of a vertex $t_i \in T$ in such a case increases proportionally to the square of n. When feasible solution components are discovered along the vertex set of G, ants located at t_i will tend to consider a set of topologies that does not encompass all topologies in its neighborhood. To do this, we propose that traversal in the space is reduced to a binary decision to move to an existing neighbor that has already been visited or to jump to a random unexplored vertex. This is a probabilistic decision influenced by the weights of existing edges and an initial weight for all of the unvisited edges that can be possibly formed.

Movement expenditure of energy Λ is calculated by Equation 1 as an ant travels along a path. More of an expenditure of energy is made if the path has not yet been explored. Otherwise, a fractional amount is expended depending on the edge weight. Notice that highly travelled paths are virtually free to traverse.

$$\Lambda_{ij} = \begin{cases} 1/w_{ij}, & \text{if existing path} \\ 1, & \text{if unvisited path} \end{cases} \tag{1}$$

This sequence of items should result in an algorithm (Algorithm 1) that is capable of scaling to large input and performing an adaptive search of the graph analogous to a breadth first search in unexplored regions of the graph, and depth first search where there is a strong pheromone concentration.

Algorithm 1: Phylogenetic Ant Colony Exploration.

```
Require: Phylogenetic Landscape $G = (T, E)$ for $n$ taxa
Require: Starting Colony Location in $T$
Require: $e \leftarrow$ Evaporation Constant, $0 < e \leq 1$
Require: max ← Maximum Number of Ant Agents
Require: init ← Starting Pheromone Concentration, init $> 0$
  ants ← Population of Ant Agents
  while still exploring do
      createAnt(ants,$n$)                ▷ Generate new ant.
      for ant in ants do                 ▷ Perform all ant movements.
          if ant.getEnergy() $> 0$ then
              moveAntForward($G$,ant)
          else
              moveAntBackward(ant)
              if ant.isDead() then
                  Remove ant from ants
              end if
          end if
      end for
      for edge in $E$ do                 ▷ Evaporate all pheromones.
          edge.reduceWeight($e$)
      end for
  end while
```

2.2 Performance Evaluation

In order to test the effect of a number of free parameters in the proposed algorithm on its to sample the space, the proposed algorithm was run on existing sequence data. The algorithm was applied to both empirical biological sequence data ($n = 23$) as well as synthetic sequence data ($n = 9$), of which the full search space can be explored, in order to compare both the difference arising from different taxon set size and simulation. Stamatakis and Albright et al. both claim that simulated phylogenetic tree data tends to be less complex than real biological data. As a result, the landscapes that result from simulated data are not entirely representative in terms of the complexity that may result from actual data (Albright et al., 2014; Stamatakis, 2014). This hypothesis will be investigated when considering the results of the experiments.

The data used for evaluation was involved in a phylogenetic examination done by Meehan et al. which was used to investigate complex phylogenetic properties of the Lachnospiraceae family (Meehan and Beiko, 2013). 16S RNA sequence information regarding a selection of species in this family was used as input for phylogenetic tree reconstruction in the proposed algorithm.

Definition Let the free parameters for the proposed method be denoted e for an evaporation constant, m for the maximum number of possible ant agents, i for the exponent of the initial pheromone concentration of edges such that 2^i is the starting weight, and t_0 corresponds to the starting topology.

After running a series of experiments, 45 landscapes were created for both the empirical and synthetic data, each with unique sets of parameter values, testing the parameters e, m and i, to thoroughly span the domain of possible values for each parameter. These were all given a starting topology that corresponds to one that scores well as found by a heuristic known as FastTree (Price et al., 2009). For a selection of the landscapes with parameters that searched the widest portion of the space, replicate runs will be performed with identical parameters and with a variation in starting location t_0. For all experiments, the number of iterations was kept fixed at 10000.

Various quantities on resultant explored graphs were recorded. These quantities include the range of fitness function values, for which we use log-likelihood, range of node degree, number of trees, and their graph diameter. Log-likelihood was calculated using the C phylogenetic likelihood library (http://www.libpll.org). Node degree refers to the number of edges connected to a given node. Another evaluation metric involves considering the number of different splits among explored trees as a measure of diversity in tree topologies. A greater number of splits, or *bipartitions* of trees, implies a greater diversity of trees found. Consideration of these properties should determine the breadth of sampling that was done across the space.

A final test that was carried out on the explored landscapes involved ranking all found trees by their log-likelihood. Then, we defined a confidence set of trees amongst the top 10% of these trees from results of the Approximately Unbiased (AU) test. The AU test is a procedure which provides a selection of trees which is most likely to to include the true tree amongst a selection of trees (Shimodaira, 2002). The AU test was applied to these selections of trees using the CONSEL application (Shimodaira and Hasegawa, 2001).

Table 1: **Replicate Run Landscape Quantities**. The σ and $\bar{x}$ noted by each quantity indicates these are standard deviations and averages of each measure respectively. Let A represent the confidence set of trees computed by the AU Test. [*] Parameters are triplets where (a) $e = 0.25, i = -8, m = 5$, (b) $e = 0.50, i = 0, m = 10$, (c) $e = 0.75, i = 0, m = 10$, and (d) $e = 0.00, i = 8, m = 10$. [†] The number of bipartitions refers to the number of unique clades or topological splits in the trees.

Data Type	Set[*]	σ Num. Bipartitions[†]	σ Max Degree	σ Avg. Log-Likelihood	$\bar{x}$ Avg. Log-Likelihood	σ Max Log-Likelihood	$\bar{x}$ $\|A\|$
Empirical $n = 23$	(a)	4726	23.7	95.7	-12384.7	138	20
	(b)	10579	104	149	-12406.4	74.5	31
	(c)	13080	210	128	-12368.7	38.9	34
Synthetic $n = 9$	(d)	0.376	9.00	17.0	-11014.7	14.6	2
	(b)	0.855	8.90	13.3	-11016.0	9.68	3
	(c)	0.519	8.67	19.3	-11016.5	21.1	3

Table 2: **Varied Starting Topologies Run Landscape Quantities**. The σ and $\bar{x}$ noted by each quantity indicates these are standard deviations and averages of each measure respectively. Let A represent the confidence set of trees computed by the AU Test. [*] Parameters are triplets where (a) $e = 0.25, i = -8, m = 5$, (b) $e = 0.50, i = 0, m = 10$, (c) $e = 0.75, i = 0, m = 10$, and (d) $e = 0.00, i = 8, m = 10$. [†] The number of bipartitions refers to the number of unique clades or topological splits in the trees.

Data Type	Set[*]	σ Num. Bipartitions[†]	σ Max Degree	σ Avg. Log-Likelihood	$\bar{x}$ Avg. Log-Likelihood	σ Max Log-Likelihood	$\bar{x}$ $\|A\|$
Empirical $n = 23$	(a)	10883	63.2	616	-12955.5	739	16
	(b)	25297	182	257	-12512.3	182	37
	(c)	16210	171	602	-12743.7	538	27
Synthetic $n = 9$	(d)	0.870	13.3	14.4	-11005.9	22.2	2
	(b)	2.140	17.3	18.2	-11014.3	14.7	3
	(c)	1.630	15.9	21.4	-11010.8	36.6	3

3 RESULTS

When investigating different sets of e, m and i parameters, it was found that the diameter for every landscape appears to remain constant regardless of how parameters were varied. The diameter of the phylogenetic space is meant to be $\theta(n)$, and the diameters found were slightly less than n. The diameter for empirical data landscapes were found to be equal to 15-16 nodes, and for the synthetic data landscapes it was 5. The fact that these values are smaller than expected is hypothesized to be due to the existence of suboptimal regions that are not reached by the PACO algorithm.

What *does* appear to differ regards the quantity and range of scores for trees visited between these different topologies. The number of trees and bipartitions found differed mostly when e and m were varied. There appears to be a significant proportionality between the number of ants and how many bipartitions are found. A greater quantity of ants implies more work being done at every iteration of the algorithm. Furthermore, these ants also *interact* with each other through the deposited pheromone applied on edges.

The evaporation constant appears to significantly affect the quality of fitness of the topologies the ant agents visit. The best collection of trees is found around a constant of $e = 0.5$, but extreme e values leads to a drop in the ability of the algorithm to explore the most relevant regions of the space. This dramatically reduces the relative fitness of found trees. Pheromone concentrations across edges effectively encode a long-term memory of ants upon the surface. It is surmised that evaporation provides the algorithm the ability to forget poor regions and reinforce the exploration of higher likelihood regions.

We selected three triplets of parameters where search properties were satisfactory and kept them fixed to test for robustness of the search. Across replicate runs with these selections (Table 1), we find similar properties of broad exploration through the space. All of the replicate runs consistently generated landscapes with a large number of trees. However, when we investigate results from the empirical data, a large deviation exists amongst the replicate runs for the number of bipartitions and the maximum degree found in the search space. The former deviation signifies variation in the algorithm's ability to find a great diversity of trees across the runs. The latter suggests inconsistent behavior when ants are causing edges to be created, possibly due to differently scored trees being visited. Despite this, the difference in log-likelihood is small and the nodes where the degree is largest are those that score higher. Therefore, while different breadths of tree diversity is being acquired, and while different trees are being visited between replicate runs, the ability of the algorithm to sample similarly scoring trees and regions does not appear to change. Notice, also, that this inference is less relevant if we discuss the synthetic dataset.

Being a smaller search space, the algorithm appears to sample the trees found in the smaller space thoroughly. Respectively, the algorithm explored a number of the possible trees in the empirical landscape on the order of $10^{-15}\%$ and of the synthetic dataset on the order of 1%. As the number of trees in the space is equal to $O(n!)$, this shows that although the algorithm is not searching a very large proportion of all possible trees, it is sampling a number of them that is sufficient to acquire a shape of the landscape.

When choosing different starting topologies (Table 2), the deviations in number of bipartitions and average log-likelihood are magnified between both empirical and synthetic data. It appears that when a different, possibly worse starting topology, is chosen, more iterations need to be done in order to acquire more of a diversity in splits and to bring resultant landscapes consistently into regions of better scoring topologies. However, when a good path is found, the energy expenditure function should mitigate this effect.

When the AU test confidence sets of trees were computed for, the number of trees found to be present in these sets were similar for respective sets of parameters and for both emperical and synthetic data. Even when starting topologies were varied, the number of trees found to be part of these sets did not seem to be reduced from those found by starting at a well-scoring topology. This suggests a tendency for the search to find well-scoring regions of trees.

4 CONCLUSION

This metaheuristic was designed to sample a large number of regions of interest of the search space with a reasonable number of iterations and amount of time. In order to acquire an understanding of its performance, a number of parameters possessed by the algorithm can be tuned. We found that evaporation was effective in steering the search to well-scoring regions of the space, the number of ant agents extended the number of trees found, and that the highest scoring trees in the search were visited more often as indicated by their increased degree.

Two rounds of experimentation were carried out including a first round testing for different triplets of parameter values. The second round of experimentation saw the investigation of replicate runs and the starting topology being varied. All results show that, when exploring both empirical and synthetic data, we can make three claims about the performance of the proposed algorithm. Firstly, the PLACO algorithm is capable of broadly exploring the combinatorial space in spite of the number of taxa. Secondly, across replicate runs we find consistent behaviour but variation in quality of trees. This implies a sparse but broad search where different topologies are being found. Thirdly, it does not matter where the algorithm starts in order to acquire a wide ranging set of trees and to sample properties of and the shape of the space.

Future work investigate the maintenance of multiple populations in the space. For example, we could build into the algorithm an ability for it to iteratively create colonies. This can accomplish to more densely move across the space and focus on regions of particular interest.

REFERENCES

Albright, E., Hessel, J., Hiranuma, N., Wang, C., and Goings, S. (2014). A comparative analysis of popular phylogenetic reconstruction algorithms. *Midwest Instruction and Computing Symposium (MICS) 2014 Proceedings*.

Bastert, O., Rockmore, D., Stadler, P. F., and Tinhofer, G. (2002). Landscapes on spaces of trees. *Applied mathematics and computation*, 131(2):439–459.

Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.*, 35(3):268–308.

Charleston, M. A. (1995). Toward a characterization of landscapes of combinatorial optimization problems, with special attention to the phylogeny problem. *Journal of Computational Biology*, 2(3):439–450.

Dorigo, M., Di Caro, G., and Gambardella, L. M. (1999). Ant algorithms for discrete optimization. *Artificial life*, 5(2):137–172.

Felsenstein, J. (2004). *Inferring phylogenies*, volume 2. Sinauer Associates Sunderland.

Fitch, W. M. (1971). Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Biology*, 20(4):406–416.

Foulds, L. R. and Graham, R. L. (1982). The steiner problem in phylogeny is np-complete. *Advances in Applied Mathematics*, 3(1):43–49.

Luke, S. (2013). *Essentials of Metaheuristics*. Lulu, second edition.

Meehan, C. J. and Beiko, R. G. (2013). A phylogenomic view of ecological specialization in the lachnospiraceae, a family of digestive tract-associated bacteria. Technical report, PeerJ PrePrints.

Price, M. N., Dehal, P. S., and Arkin, A. P. (2009). Fasttree: computing large minimum evolution trees with profiles instead of a distance matrix. *Molecular biology and evolution*, 26(7):1641–1650.

Sanderson, M. J., McMahon, M. M., and Steel, M. (2011). Terraces in phylogenetic tree space. *Science*, 333(6041):448–450.

Shimodaira, H. (2002). An approximately unbiased test of phylogenetic tree selection. *Systematic Biology*, 51(3):492–508.

Shimodaira, H. and Hasegawa, M. (2001). Consel: for assessing the confidence of phylogenetic tree selection. *Bioinformatics*, 17(12):1246–1247.

Stadler, P. F. (1996). Landscapes and their correlation functions. *Journal of Mathematical chemistry*, 20(1):1–45.

Stamatakis, A. (2014). Raxml version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*.