Review of Phylogenetic Tree Construction

Jeffrey Rizzo[1] and Eric C. Rouchka[1]

TR-ULBL-2007-01

November 19, 2007

[1]University of Louisville
Speed School of Engineering
Department of Computer Engineering and Computer Science
123 JB Speed Building
Louisville, Kenucky, USA 40292

jvrizzo@gmail.com; eric.rouchka@louisville.edu

UNIVERSITY of LOUISVILLE dare to be great | Bioinformatics Laboratory Technical Report Series

Bioinformatics Review

Review of Phylogenetic Tree Construction

Jeffrey Rizzo[1] and Eric C. Rouchka[1,*]

[1]Department of Computer Engineering and Computer Science, University of Louisville, 123 JB Speed Building, Louisville, KY, USA

UNIVERSITY OF LOUISVILLE BIOINFORMATICS LABORATORY TECHNICAL REPORT SERIES REPORT NUMBER **TR-ULBL-2007-01**

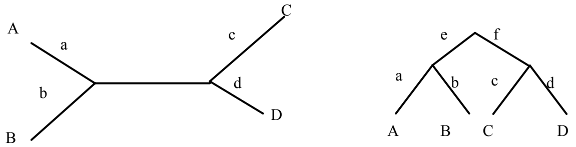
ABSTRACT

Motivation: Phylogenetic tree construction is a complex yet important problem in the field of bioinformatics. Once constructed, a phylogenetic or evolutionary tree can lend insight into the evolution of different species. The issue is that for a large number of species the problem grows to a computational complexity that is not easily solved. For this reason, new methods are being researched and applied to phylogenetic tree construction and have provided some promising results. Two topics of interest for this paper are the use of Ant Colony Optimization and Particle Swarm Optimization both of which are based on algorithms discovered from studding the patterns of nature.

1 INTRODUCTION

To begin, phylogenetics is the science which studies evolutionary relationship between species. In order to make predictions about these relationships, phylogenetic trees are constructed which link the species. The problem of phylogenetic tree construction is widely accepted as an area in need of more research in bioinformatics as it is considered to be an NP-complete problem which means it is in a very small class of the most difficult problems to solve [1].

A relationship between two species is classified as a phylogeny. A phylogenetic tree is a binary tree representation of the resulting relationship. There are two main types of trees that can be found: 1) rooted trees -- those that have a single node from which all nodes are derived, and 2) unrooted trees -- those that do not originate from one clear node. The tree follows standard graph theory notation were each species is represented as a node or leaf, and the relationship between species is referred to as an edge or branch. The lengths of the branches represent the time estimate between the species. A simple representation of this is illustrated in Figure 1.



(a) **(b)**

Fig. 1 (a) unrooted tree, (b) rooted tree

Table 1 illustrates the complexity of enumerating possible tree configurations by showing the number of rooted and unrooted trees based on the number operational taxonomic units (OTU). The OTU is an extant present at an external node or leaf; which in the context of graph theory is just the nodes. The table is formulated using Equation 1 for the number of unrooted trees, and Equation 2 for the number of rooted trees. Having noted that there is a large number of possible trees, it is important to distinguish that there is only one "true" or correct tree from which species have evolved. Thus finding the one correct tree can become a computational nightmare without an efficient algorithm or strategy.

$$N_U = \frac{(2n-5)!}{2^{n-3}(n-3)!} \qquad \text{(Eq. 1)}$$

$$N_R = \frac{(2n-3)!}{2^{n-2}\ (n-2)!} \qquad \text{(Eq. 2)}$$

Table 1. Number of phylogenetic trees. (Adapted from [2]).

# of OTUs	Number of Rooted	Number of Unrooted
3	3	1
4	15	3
…		
10	34,489,707	2,027,025
15	213,458,046,676,875	8 x 10^{12}
20	8 x 10^{21}	2 x10^{20}
50	2.8 x 10^{76}	3 x 20^{74}

2 METHODS

Phylogenetic tree construction methods are widely accepted to fall into one of two categories: distance based and character based. These two categories both offer a vast variety of options when constructing trees in two different directions. The most common distance based methods are the unwieghted pair group method using arithmetic averages (UPGMA) [3], Neighbor Joining [4] and the Fitch and Margoliash [5] algorithms that are all based off the initial creation of a distance matrix. The alternative to these methods is the character based methods such as maximum parsimony [6] and maximum likelihood [7] which take a probabilistic approach to tree construction. An attempt to introduce these topics

[*]To whom correspondence should be addressed.

and new approaches such as Ant Colony Optimization and Particle Optimization proceeds in the following sections.

2.1 Common Approaches

2.1.1. UPGMA and Neighbor Joining

UPGMA and Neighbor Joining use a clustering procedure that is commonly found in data mining techniques. The method is simple and intuitive [8] which makes it appealing. The method works by clustering nodes at each stage and then forming a new node on a tree. This process continues from the bottom of the tree and in each step a new node is added, and the tree grows upward. The length of the branch at each step is determined by the difference in heights of the nodes at each end of the branch. UPGMA has built in assumptions that the tree is additive and that all nodes are equally distance from the root. Since a "molecular clock" hypothesis assumption poses biological issues, UPGMA is not used much today, but gave way to a very common approach now termed "Neighbor Joining" [4].

Neighbor Joining (NJ) works like UPGMA in that it creates a new distance matrix at each step, and creates the tree based on the matrices. The difference is that NJ does not construct clusters but directly calculates distances to internal nodes. The first step in the NJ algorithm is to create a matrix with the Hamming distance between each node or taxa. The minimal distance is then used to calculate the distance from the two nodes to the node that directly links them. From there, a new matrix is calculated and the new node is substituted for the original nodes that are now joined. The advantage here is that there is not an assumption about the distances between nodes since it is directly calculated.

```
Neighbor Joining Algorithm
Initialization:
    • Define T to be the set of leaf nodes, one for
      each given sequence and put L=T.
Iteration:
    • Pick a pair of $i, j$ in L for which distance
      from $i$ to $j$ is minimal.
    • Define a new node k and set $d_{km} = ½ (d_{im} +
      d_{jm} + d_{ij})$, for all M in L.
    • Add k to T with edges of lengths $d_{ik}= ½ (d_{ij}
      + r_i – r_j)$, $d_{jk} = d_{ij} - d_{ik}$ joining $k$ to $i$ and $j$ re-
      spectively.
    • Remove $i$ and $j$ from L and add $k$.
Termination:
    • When L consists of two leaves $i$ and $j$ and
      the remaining edges between $i$ and $j$, with
      length $d_{ij}$.
```

Algorithm 1. Neighbor Joining [8].

Additivity is a measurement that depends on the distance measure used. Neighbor Joining works even if the lengths are not additive but the tree is no longer guaranteed to be the correct tree. There exists a four-point condition that can be used to test for additivity. A result of additivity is that the sum of the lengths of two distances must be greater than the third distance. For example, let d_{ij} represent the distance from i to j. If four nodes exist, then $d_{ij} + d_{kl} = d_{ik} + d_{jl}$ and is greater than $d_{il} + d_{jk}$. This is because the inclusion of the link between the two smaller clusters is common in two of the distance sums.

Since The Neighbor Joining approach to tree construction takes advantage of common clustering techniques, is it efficient to execute and easy to understand. It produces an unrooted tree that shows the relationship between sequences without assigning a root node from which all other sequences have been derived. In order to construct a tree with a common ancestor node, an outgroup species is chosen that is distantly related to the remaining sequences. The location where the new species connects to the recently constructed tree is a good indicator for the most likely location for the root of the tree. If it is not easy or possible to find an outgroup, other strategies allow one to locate a root of the tree such as using the midpoint of the longest chain of consecutive edges, which would indicate the root if the tree followed the molecular clock within reason [8].

2.1.2. Maximum Parsimony

Maximum Parsiomony (MP) is probably the most widely and accepted method of tree construction to date [8]. The method is different from the previously discussed distance based methods since it uses a character based algorithm. The method works by searching through possible tree structures and assigning a cost to each tree. Parsimony is based on the assumption that the mostly likely tree is the one that requires the fewest number of changes to explain the data in the alignment [9]. The premise that taxa or nodes sharing a common characteristic do so because the inherited that characteristics from a common ancestor.

Conflicts with this major assumption are explained under the term *homoplasy.* There are three main ways to reserve conflicts: reversal (revert back to original state), convergence (unrelated taxa evolved the same characteristic completely independently) and parallelism (different taxa may have similar mechanisms that cause a characteristic to develop in a certain manner) [9]. The tree with the lowest tree score or length, as defined by the number of changes summed along the branches, becomes the most parsimonious tree and is taken as the tree that best represents the evolutionary pattern [10].

Maximum Parsimony is also different from the other methods in that it does not find branch lengths but rather the total overall length in terms of the number of changes. Often MP, finds two or more trees that it deems equal and does not provide a definite answer in how to distinguish which tree represents the actual evolutionary tree. In most cases a strict (majority rule) consensus is used to solve this dilemma [10].

Traditional parsimony uses recursion to search for the minimal number of changes within the trees. This done by starting at the leaf of a tree and working up towards the root, which is known as post-order traversal [8]. Another version of parsimony, weighted parsimony, adds a cost factor to the algorithm and weights certain scenarios accordingly.

An artifact called long-branch attraction sometimes occurs in parsimony and should be handled. The branch length indicates the number of substitutions between two taxa or nodes. Parsimony assumes that all taxa evolve at the same rate and contribute that same amount of information. Long-branch is the phenomenon in which rapidly evolving taxa are placed together on a tree because they have many mutations. Anytime two long branches are present, they may be attracted to one another [2].

2.1.3. Maximum Likelihood

Proposed in 1981 by Felsenstein [7], Maximum likelihood (ML) is among the most computationally intensive approach but is also the most flexible [10]. ML optimizes the likelihood of observing the data given a tree topology and a model of nucleotide evolution [10]. Maximum Likelihood finds the tree that explains the observed data with the greatest probability under a specific model of evolution. ML is different from the other methods in that it is based on probability. The probability or likelihood of a tree can be computed using a few equations. Suppose a tree T with edge lengths t. Let α(*i*) denote the immediate ancestral node to *i.* Let [illegible] denote the residues at the u^{th} site of the *n* sequences. The probability of generating these residues at the *n* leaves of T is given by multiplying the probability of substitutions at all edges of the tree. The Likelihood, L = P(data | tree) which is the probably of observing the data given the tree [10].

One of the big advantages to ML is the ability to make statistical comparisons between topologies and data sets. ML can return several equally likely trees – a pro and con depending on the study [10]. Maximum Likelihood makes assumptions that the model used is accurate and if the model does not accurately reflect the underlying data set, the method is inconsistent. ML is designed to be robust, but breaching is assumptions can cause problems.

A disadvantage of ML is the extensive computation as well as new evidence that suggests there can be multiple maximum likelihood points for a given phylogenetic tree [11].

2.2 Alternative Methods

In recent years the extreme complexity of phylogenetic tree construction has prompted researchers to look to new methods. With the development of optimization techniques in the field of Artificial Intelligence, these new methods have come in two new algorithms, Ant Colony Optimization (ACO) and Particle Swarm Optimization (PCO).

2.2.1. Ant Colony Optimization

The idea to use insects to solve problems arose from the study of the behavior of termites in the 1940's and 50's by a French entomologist Pierre-Paul Grasse who observed that some species of termites reach to what he later termed "stigmergy" [1]. He defined stigmergy as a particular type of communication that termites use to communicate by modifying the environment. Stigmergy has two main aspects: it is an indirect, non-symbolic form of communication mediated by the environment and that is local and can only be accessed by those insects who visit the immediate neighborhood where it was released.

The concept of stigmergy led to the discovery that ants deposit a substance on the ground as they move to and from the food source known as pheromone. This pheromone is then used by other ants whom will follow the path with the highest amount of pheromone. This allows the ants to optimize the path from the food to the nest.

As shown in Fig. 2, ants have the ability to overcome an obstacle and find the most efficient way to return to their nest. The obstacle present in B, causes the ants to find two new paths (Fig. 2. C) and eventually the pheromone trail of the shorter path surpasses the level of pheromone on the longer trail, and the ants begin to travel the shorter path based on the communication from the other ants (Fig. 2. D)

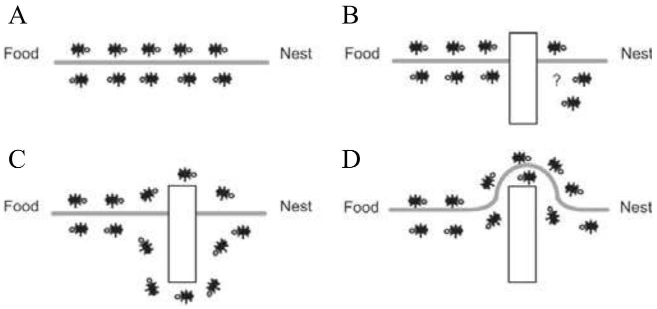


Fig. 2. Ants avoiding obstacles (from Nakhleh et al., [12]).

The algorithm written to reflect this biological phenomenon was developed in 1991 by Dorigo and was named Ant System (AS) [11]. The algorithm was first used to solve the classic Traveling Salesman Problem (TSP). ACO was found to produce relatively good results. The question remained whether or not ACO could converge to an optimal solution. While it is difficult to prove convergence in all cases due to the variations between different ACO algorithms, it has been proved that the two most common approaches converge [1;11]. The only remaining unanswered question is the amount of time and resources that it will take to solve the problem.

ACO has been developed into a metaheuristic. The goal of ACO is to minimize the solution, however it is easy to see that minimizing one result can be the same as maximizing the inverse and therefore ACO can be applied to maximization problems too. In order for the ant colony algorithm to work in all optimization problems, Algorithm 2 was devised in a very general form that can be modified for different applications.

The ACO Metaheuristic

```
Set parameters, initialize pheromone trails
while termination condition not met do
        ConstructAntSolutions
        ApplyLocalSearch (optional)
        UpdatePheromones
endwhile
```

Algorithm 2. Ant Colony Optimization. (from [1]).

The three phases of the search are briefly described below. *Constructing Ant Solutions* refers to the creation of the possible walks or paths that an ant can take from node *i* to all other nodes. The choice that the ant makes is typically guided by a stochastic heuristic that favors the pheromone levels. The heuristic is thus used to pick the path with the highest pheromone level. *Apply Local Search* is an optional step that is highly customized to the application, but allows for a local search to be performed before updating the pheromone. *Update Pheromones* updates the levels of the pheromone based on two conditions. The first is that pheromone levels must evaporate (decrease) with respect to time, and secondly, pheromone levels of "good" paths should be increased. These three steps are completed for the duration of the algorithm which terminates upon exploring all paths or reaching the optimal.

With a working knowledge of ACO, researchers in the bioinformatics field have looked to ACO to solve problems including those of evolutionary tree constructions. Mauricion Perretto and Heiton Silverio Lopes have introduced an adaptation of ACO that works to construct the optimal phylogenetic tree [13]. To begin, their approach creates a fully connected graph of the spices and constructs a distance matrix that is used as the distances between each node. From there the problem is solved much like TSP where the ants begin at random and explore the search space of the graph updating the edges based on the heuristics. The transition found (Eq. 3) gives the probability that the k-th ant, beginning at node *i*, goes to node *j* in its next step. $P_k(i,j)$ is the probability of transition from node *i* to *j*, τ is the pheromone trail between two nodes, *d(i,j)* is the evolutionary distance, J_i^k is the set of nodes connected to node *i*, and already visited by the *k-th* ant and α and β are constants.

$$P_k(i,j) = \frac{[\tau(i,j)]^{\alpha}.[1/d(i,j)]^{-\beta}}{\sum_{u \in J_i^k}\left([\tau(i,u)]^{\alpha}.[1/d(i,j)]^{-\beta}\right)} \qquad \text{(Eq. 3)}$$

The equation has two main components which allow it apply the evolutionary distance as well as the experience of the ants. First the distance between the two nodes and second the accumulated pheromone levels. Since τ is dynamically updated, it represents the "attractiveness" of node *j* when the ant is at node *i.* The ant thus picks a path that minimizes that transition function, and finds the shortest evolutionary distance between two species. To differ from traditional ACO, the system presented uses an intermediary node created between the two previously visited nodes. It represents the ancestor of the two nodes, is not in the list of nodes to be set in the tree and represented by *u*. It is used to help recalculate distances to the remaining nodes by Eq. 4.

$$d_{nu}(i,j) = \begin{cases} d(i,u)+[d(i,u)-d(j,u)]\eta, & \text{if } \; d(j,u) \geq d(i,u) \\ d(j,u)+[d(j,u)-d(i,u)]\eta, & \text{if } \; d(i,u) > d(j,u) \end{cases} \qquad \text{(Eq. 4)}$$

The above steps repeat until all nodes are added to the list of visited nodes and then a final path is found. The final distance or score of the path is calculated by summing the transition probabilities along the path of adjacent nodes. When the path is constructed, the pheromone level is increment for all nodes belonging to the path and decremented for the evaporation affect. The pheromone trail matrix is updated according to Eq. 5 where ρ represents the evaporation rate of the pheromone and *Δτ(i,j)* is the rate of increment of pheromone obtained from Eq. 6.

$$\tau(i,j) = \rho \cdot \tau(i,j) + (1-\rho) \cdot \Delta\tau(i,j) \qquad \text{(Eq. 5)}$$

$$\Delta\tau(i,j) = \begin{cases} \sum_{t=0}^{k} S_{c(t)} \cdot (S_{best})^{-1} & \text{, if } \; (i,j) \in c(t) \\ 0 & \text{, otherwise} \end{cases} \qquad \text{(Eq. 6)}$$

These equations allow for the determination of the best path up to the point of the calculation. Using the procedure, each cycle moves closer toward constructing a tree. Once the correct number of cycles is completed, it is now possible to use the best path found to create the tree. The ACO has thus provided a linear list of the best possible path, and now Algorithm 3, must be used to create the tree.

```
Tree Construction Algorithm

WHILE NOT (all species grouped)
  FIND i, j pair that have the largest value in the
      pheromone matrix.
  IF (i OR j) already grouped change index by
      group index
  GROUP i, j pair into a new species k
  COMPUTE the distance between current species
      and ancestor
  DELETE the value of i, j pair
END
```

Algorithm 3. **Tree construction algorithm** (from [13].)

Perretto and Lopes found that their method was successful when compared with other more common methods. They completed tests using mtDNA from 20 different species of mammals. The results of the phylogenetic trees were compared to the common PHYLIP software package using two of the common algorithms including Neighbor Joining and Fitch and Margoliash. To compare the different trees they used the tree structure and the total distance between nodes (Eq. 7) that was proposed by Kumnorkaew et al. [14].

$$d_t = 110 + \sum_{i=0}^{n}\sum_{j=0}^{n} \frac{d_{obs}(i,j)^{2}}{d_{\exp}(i,j)} \qquad \textbf{(Eq. 7)}$$

This distance measure is similar to the quadratic error computation. The trees that were obtained with the mtDNA are shown if Fig. 3. using ACO (A) and Neighbor Joining (B). As shown, the two trees are similar, but there are small discrepancies between them which create differences in the distances between species. Table 2 shows the computed distance totals between branches using the different methods. The differences in the distances are due to the difference in the tree groupings as seen in Fig. 3.

Table 2. Distance Comparisons between methods. (Taken from [13].)

Algorithm	**Distance**
ACO	351.56
Fitch and Margoliash	352.27
Neighbor Joining	354.23

These preliminary results provide evidence that ACO can successfully be used to create phylogenetic trees and more importantly that ACO may provide better results than methods. The method has a number of parameters that are used throughout the above equations including α and β which were experimented with but were not calculated to be optimal. For this reason the evidence is only preliminary and cannot be considered as proof

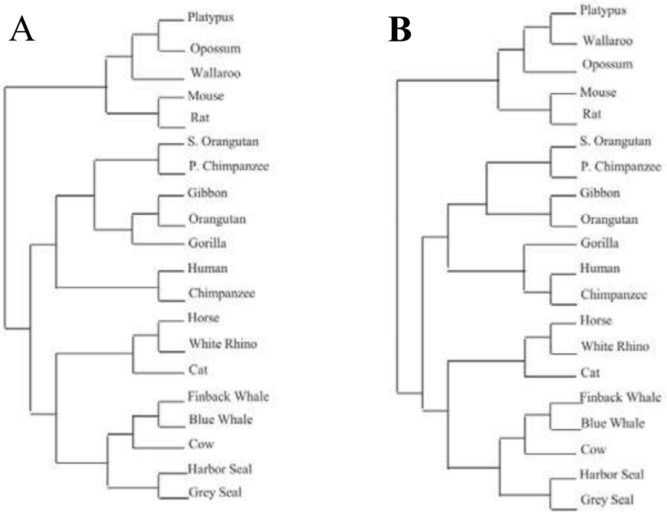
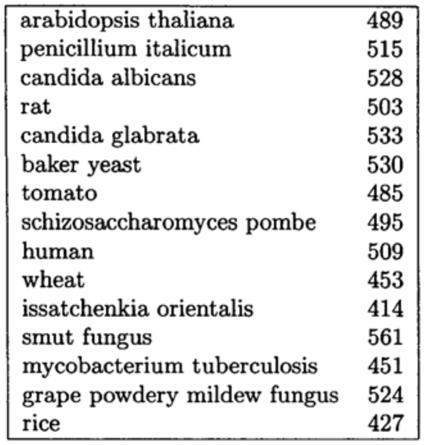


Fig. 3. Trees produced using ACO (A) and Neighbor Joining (B) Methods. [12].

that ACO is always more accurate. With these parameters comes room for improvement of the ACO algorithm presented. If these parameters can be optimized it they could actually improve the overall efficiency and accuracy of the ACO method presented. The results are thus considered promising.

ACO has also been used for other experiments [15]. Here the approach of ACO is slightly different than that discussed above, but the result is the same. When conducting experimentation, the team generated sequences using Seq-Gen which allowed them to know the correct tree for the set of sequences used in their simulation. They then used the distance matrix between the sequences as computed by PHYLIP, an online software program. This input was then fed into FITCH and Neighbor Joining algorithms on PHYLIP for comparison to their proposed ACO method. They used to different size trees with 8 and 16 leaves, respectively. For the smaller 8-leaf tree the ant algorithm gave the best general result [16]. However, for 16 leaves FITCH gave the best result. More simulation should be completed on larger trees, but the conclusion is made that ACO has proved comparable to traditional methods.



arabidopsis thaliana	489
penicillium italicum	515
candida albicans	528
rat	503
candida glabrata	533
baker yeast	530
tomato	485
schizosaccharomyces pombe	495
human	509
wheat	453
issatchenkia orientalis	414
smut fungus	561
mycobacterium tuberculosis	451
grape powdery mildew fungus	524
rice	427

Fig. 4. 15-Species and length of their CYP Sequences [16].

The same group also tested with an example protein family of the Cytochrome P450 CYP050A. The sequences of 15 species were taken with lengths varying from 414 to 561 (Fig. 4). The Multiple Sequence Alignment was obtained using ClustalW and the distance matrix computed by PHYLIP. Here all three methods (FITCH, Neighbor Joining and ACO) agreed on the tree, however the ant algorithm gave the best score of 4023.8 [16].

Another approach to ACO was presented in [17]. Here the process of constructing the tree uses a pheromone graph where the pheromone graph is adaptively updated according to the pheromone level left by the ants. The presented algorithm then dynamically adjusts the influence of each ant to the trail information updating and the selected probability of the path according to the equilibrium of the ant distribution.

The three main steps of the phylogenetic tree construction are: initialization, construction based on optimal path found by ants, and lastly optimization. The method designates the input species as cities in the TSP, and in the first step, constructs a fully connected graph using the distance matrix among the species. Nodes represent the species and the edges or distances between the cities represent the calculated distance between the species.

In constructing the tree, ants are sent out to find the optimal path by starting each ant at a randomly selected node or city. ACO theory has each ant pick its next destination based on the probability function used to describe the graph. This algorithm adjusts the equation to determine probability based on the equilibrium of path distribution dynamically. Every ant must pass through each city and all nodes must be stored in the open list, or the list of already visited nodes.

It is also necessary to obtain the evolutionary distance between two species based on one of several methods. In this algorithm, the cosine distance is introduced where the two species in the data set are represented as vectors and the cosine of the angle between them defines the evolutionary distance. It is also necessary to measure the distribution of the solutions and the trail updating information. The "gathered degree" method was used to measure this and it allowed the algorithm to determine probability of each path dynamically according to the ant distribution. The gathered degree (Eq. 8) thus determines the paths that the ants can take from each node. Ants thus consider paths with highest trail information (pheromone level). The number of paths to consider must be limited so an equation is introduced that helps limit local optimization and to ensure that paths with highest trail information are considered with highest probability (Eq. 9). This equation for *w* paths also allows ants to only consider paths with highest trail information in their selection process. The general principle is that paths with more trail information and small local distances are selected in a larger probability.

$$gd(i) = \sqrt{\sum_{j=1}^{r}\left[\frac{m}{r} - a_{ij}\right]^2} \qquad \text{(Eq. 8)}$$

$$w = \left\lfloor \frac{gd(i)}{\max gd}(r-1) + 0.5 \right\rfloor + 1 \qquad \text{(Eq. 9)}$$

Once the ants have completed their "work," the construction of the tree is simple. The pheromone matrix that was given by the ants through the TSP optimization is used to construct the tree and

```
Algorithm: Tree construction algorithm
Step1: initialization
    1.1 initializes parameters and the pheromone graph;
    1.2 For each ant do choose an initial node to visit randomly; End for
Step2: iterative process// finding the optimal path
while (not terminal conditions) do
            {2.1 For i=1 to n do // n is the number of nodes in the graph
            compute the gathered degree of node
            i and the distribution extent w of
            the ants in this iteration according
            to the equations (8) and (9)
            For k=1 to m do // m is the number of ants
            2.1.1 Select the next node j to visit.
            2.1.2 Update the trail information on
            path (i, j) locally, according
            to equation (11)
            End for k
                If in m ants, the total lengths of paths that some ant traveled currently has exceeded the length of the optimal path obtained in last iteration, then stop the iteration of this ant.
            2.2 update the trail information globally for each path of all
            sequences;
End while
Step3. Tree construction
            While not (all species grouped)
            Find a pair of species i, j that
            have the largest value in the
            pheromone matrix
            If (i or j) already grouped CHANGE
            index by group index
            Group i, j pair into a new species k;
            Compute the distance between
            Current species and ancestor;
            Delete the value of i, j pair
            End
      }
}
```

Algorithm 4. Tree Construction Algorithm. From [12].

the similarity between objects by pheromone levels on the edge of the graph. Essentially ants select the next node by the finding the edge with the largest amount of pheromone. The full algorithm used is shown in Algorithm 4.

Experiments were completed using 14 species on the proteins hemoglobin alpha-I and cyctochrome C. The results were compared to the neighbor joining method (Clustal X) and TSP approach. There were significant differences in the tree configurations but quantitatively the differences are small. The proposed method using ACO provided shorter branch lengths in all of the experiments run. It is stated that there are 94.11% cases better than TSP-Approach with basic parameters set. The conclusion is that the experiments show higher quality results than other algorithms and the authors suggest that more studies should be done in this area of phylogenetic tree construction using Ant Colony theory.

From the three different ACO experiments presented above that have been completed, it is obvious that ACO has a purpose in phylogenetic tree construction. However, more work needs to be completed. All three groups introduced the problem and gave viable solutions that proved to be successful in their tests yet each group also noted that their tests were not completely conclusive. More studies should be done and more experimentation needs to occur, but these initial results are promising.

2.2.2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a method to solve complex optimization problems using a different result from nature. PSO was originally developed in 1995 by Kennedy and Eberhart [18] and it belongs to the category of swarm intelligence methods. The idea is rooted in the study and action simulation of simplified social models of birds and schools of fish. PSO is a population-based algorithm that exploits a population of individuals to probe promising regions of a search space. Unlike ACO where the ant colonies were found to use a concept known as stigmergy to communicate, birds were found to use a communication strategy that was similar to a broadcast. Each agent is a particle like structure with the following parts: coordinates of the current location in the optimization landscape, the best solution point visited so far and the subset of other agents seen as neighbors [11].

As shown in the algorithm, during each iteration, each particle accelerates in the direction of its own person best solution found so far, as well as in the direction of the global best position discovered so far [18]. What this means is that when one particle discovers something in the right direction, it moves that direction and all others follow it. Essentially one bird moves in a certain direction and the rest of the flock move their position relative to it in order to maintain their position. An effective algorithm thus has the daunting task of balancing the individual influence with the swarm influence and effectively leading the general movement of the swarm.

The equations that are used to update the particles position and speed consist of three parts. First part is the previous speed of the swarm; the second is cognition modal (thought of the swarm); third is the social modal [1]. The first gives the swarm balance, the second allows the swarm to search the whole and avoid issues with locality, and the third reflects the information that is broadcasted out or communicated within the swarm.

The experimentation and development of “Discrete Particle Swarm Optimization” for phylogenetic tree construction gave results that showed PSO is well suited for cases when the number of sequences is less than 40. This was determined after analysis that the PSO is able to converge quickly on the best tree compared with genetic algorithm, but if the number of sequences was too large the efficiency will decrease [1].

3 RESULTS

The results of the following sections give some evidence that phylogenetic tree construction is a complex problem that is difficult to solve. By providing new methods of optimizing and searching for the best tree, some of the traditional methods are challenged and reviewed closely for their relevance and acceptance. ACO and PSO both have been proven only in small studies to be more efficient and/or accurate than traditional methods such as Neighbor Joining and Maximum Likelihood, but these small studies are only a step toward the research that will be done in the future. As more and more sequences become readily available, the need to construct evolutionary trees will continue to grow and new methods for doing so will be sought.

Ant Colony Optimization seems to be the most promising of the two methods explored within this paper, mainly because more research has been conducted in that direction. It appears that more researchers are using Ant Colony as the optimal method within the branch of Swarm Intelligence that includes ACO and PSO as the two forerunners. Perhaps the development of ACO about five years prior to PSO has provided the algorithm with more accep-

```
procedure Particle Swarm Optimization()
    foreach particle in ParticleSet do
        init at random positions and velocity;
        select at random the neighbor set;
    end foreach
    while (¬ stopping criterion)
        foreach particle in ParticleSet do
            calculate current fitness and update memory;
            get neighbor with best fitness;
            calculate individual deviation between current and
            best so far fitness;
            calculate social deviation between current and best
            neighbor fitness;
            calculate velocity vector variation as weighted sum
            between deviations;
            update velocity vector;
        end foreach
    end while
return best solution generated;
```

Algorithm 5. **Particle Swarm Optimization**. From [5]

tances by the research communities. As shown within, ACO is a new approach to phylogenetic tree construction and will continue to grow and possible become accepted as the standard over FITCH, Neighbor Joining and Maximum Likelihood someday. The initial results are promising and researchers need only to perfect their algorithms for all cases and ACO just may emerge as the optimal method for phylogenetic tree construction.

4 ACKNOWLEDGEMENTS

This majority of this paper was written by JR to fulfill a requirement for CECS 660 Introduction to Bioinformatics during the summer of 2007. ER is supported by NIH-NCRR grant P20RR16481 and NIH-NIEHS grant P30ES014443-01A1. The contents of this manuscript are solely the responsibility of the authors and do not represent the official views of NCRR, NIEHS, or NIH.

REFERENCES

[1] H.-Y. LV, W.-G. Zhou, and C.-G. Zhou, "A Discrete Particle Swarm Optimization Algorithm for Phylogenetic Tree Reconstruction," 2004, pp. 2650-2654.

[2] J. Pevsner, "Molecular Phylogeny and Evolution,", 1 ed John Wiley and Sons, Inc., 2003, pp. 368-388.

[3] Sneath PHA and Sokal RR, "Numerical Taxonomy," in *Numerical Taxonomy* San Francisco, CA: Freeman, 1973, pp. 230-234.

[4] N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," *Mol. Biol. Evol.*, vol. 4, no. 4, pp. 406-425, July1987.

[5] W. M. Fitch and E. Margoliash, "Construction of phylogenetic trees," *Science*, vol. 155, no. 760, pp. 279-284, Jan.1967.

[6] W. Fitch, "Toward defining the course of evolution: minimum change for a specified tree topology.," *Syst. Zool.*, vol. 20, pp. 406-416, Jan.1971.

[7] J. Felsenstein, "Evolutionary trees from DNA sequences: a maximum likelihood approach," *J. Mol. Evol.*, vol. 17, no. 6, pp. 368-376, 1981.

[8] Durbin R., S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Reprint ed. Cambridge: Cambridge University Press, 1999.

[9] B. G. Hall, *Phylogenetic Trees Made Easy: A How-to Manual*, 3 ed Sinauer Associates, Inc., 2007.

[10] A. N. Egan and K. A. Crandall, "Theory of Phylogenetic Estimation," in *Evolutionary Genetics: Concepts and Case Studies,* 1 ed Oxford University Press, USA, 2006, pp. 426-436.

[11] M. Dorigo and G. Di Caro, "The Ant Colony Optimization Meta-Heuristic," in *New Ideas in Optimization*. Corne D., M. Dorigo, and F. Glover, Eds. London: McGraw-Hill, 1999, pp. 11-32.

[12] L. Nakhleh, G. Jin, F. Zhao, and J. Mellor-Crummey, "Reconstructing phylogenetic networks using maximum parsimony," *Proc. IEEE Comput. Syst. Bioinform. Conf.*, pp. 93-102, 2005.

[13] M. Perretto and H. S. Lopes, "Reconstruction of phylogenetic trees using the ant colony optimization algorithm," *Genet. Mol. Res.*, vol. 4, no. 3, pp. 581-589, 2005.

[14] C. Kumnorkaew, K. Ku, and P. Ruenglertpanyakul, "Application of ant colony optimization to evolutionary tree construction," 2004.

[15] S. Ando and H. Iba, "Ant algorithm for construction of evolutionary tree," Honolulu, HI, USA: 2007, pp. 1552-1557.

[16] M. Dorigo and T. Stutzle, *Ant Colony Optimization* Mit Press, 2004.

[17] J. Guo, L. Chen, L. Qin, and C. Wang, "A self-adaptive ant colony algorithm for phylogenetic tree construction,", 01 ed 2006, p. 78.

[18] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," Piscataway, NJ, USA: IEEE Service Center, 1995, pp. 1942-1948.