Journal of Intelligent & Fuzzy Systems 18 (2007) 1–9
IOS Press

An Ant Colony system for large-scale phylogenetic tree reconstruction

Heitor S. Lopes* and Mauricio Perretto[1]
Laboratório de Bioinformática / CPGEI, Universidade Tecnológica Federal do Paraná, UTFPR, Av. 7 de setembro, 3165, 80230-901 Curitiba (PR), Brazil

Abstract. An important problem in Bioinformatics is the reconstruction of phylogenetic trees. A phylogenetic tree aims at unveiling the evolutionary relationship between several species. In this way, it is possible to know which species are more closely related to one another and which are more distantly related. Established methods for phylogeny work fine for small or moderate number of species, but they become unfeasible for large-scale phylogeny. This work proposes a methodology using the Ant Colony Optimization (ACO) paradigm for the problem. A phylogenetic tree is viewed as a fully-connected graph using a matrix of distances between species. We search for the shortest path in this graph, turning the problem to an instance of the well-known traveling salesman problem. After, we describe how to build a tree using the directed graph and the pheromone matrix obtained by the ACO. Two data sets were used to test the system. The first one was used to investigate the sensitivity of the control parameters and to define their default values. The second data set was used to analyze the scalability of the system for a large number of sequences. Results show that the proposed method is as good as or even better than the other conventional methods and very efficient for large-scale phylogeny.

Keywords: Bioinformatics, phylogenetic tree, Ant Colony Optimization

1. Introduction

1.1. Phylogenetic trees

The evolutionary relationship among different species can be accessed by means of phylogenetic trees. Such assessment tries to unveil how these species might have been derived during the evolution of live on earth. This can be done by analyzing a set of DNA (or amino acids) sequences from different species. The construction of phylogenetic trees is an important problem in Bioinformatics and, like many others, it is still an open subject for research. This is mainly due to the NP complexity of the problem [8] that leads to intractable search spaces when dealing with the phylogeny of a large number of species.

In a simple way, a phylogenetic tree can be considered a binary tree, whose leaf nodes represent the species to be analyzed and inner nodes are the ancestral species from which they have evolved. Phylogenetic trees can have or not a root (see Fig. 1) that indicates the oldest ancestral. Usually, a rooted tree represents better the phylogenetic history of species. On the other hand, an unrooted tree represents better the possible correlation between species.

Equation (1) shows how many different trees (NT) can be computed for both, rooted and unrooted trees, using n species [5]. For instance, if we would like to find the best unrooted tree using the method of maximum similarity for (only) 20 species, we should try 8,200,794,532,637,891,559,375 trees. This simple example shows how fast the problem becomes intractable as the number of species increase. Therefore, a number of heuristic methods have been proposed for the reconstruction of phylogenetic trees, such as genetic

[1] M. Perretto is currently with Departamento de Engenharia da Computação, Centro Universitário Positivo, UNICENP, Curitiba (PR), Brazil.

*Corresponding author. Tel.: +55 41 3310 4694; E-mail: hslopes @pesquisador.cnpq.br.

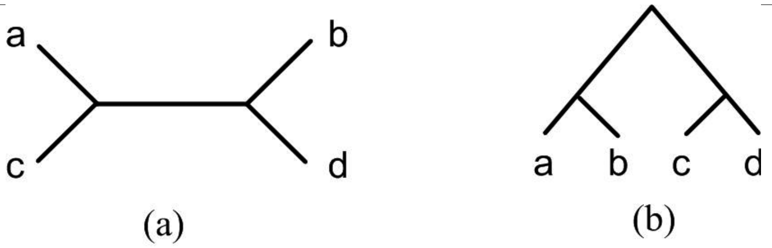

Fig. 1. Topologies of phylogenetic trees: (a) unrooted tree, (b) rooted tree.

algorithms [12] and simulated annealing [18].

$$NT = \begin{cases} \frac{(2n-3)!}{2^{(n-2)}(n-2)!} \text{ for unrooted trees} \\ \\ \frac{(2n-5)!}{2^{(n-3)}(n-3)!} \text{ for rooted trees} \end{cases} \qquad (1)$$

Although there is still no consensus, the current methods for the reconstruction of phylogenetic trees can be roughly grouped into two families: feature-based methods and distance-based methods [9].

Feature-based methods comprise both parsimony and probabilistic methods. Parsimony methods are based on the principle that correct phylogenetic trees are those that encompasses the smallest number of evolutionary changes among species. These methods are computationally expensive and sensitive to misalignment errors. Probabilistic methods, such as the Maximum Likelihood proposed by Felsenstein [5], are known to be the most robust to input errors and to produce better results. However, they have serious limitations. Since they work directly with pre-aligned sequences, the multiple-sequence alignment algorithm used can introduce errors and, besides, it is also computationally expensive for a large number of sequences.

On the other hand, distance-based methods are based on the principle of similarity. Examples of these methods are UPGMA (Unweighted Pair Group Method using arithmetic Averages) [19] and Neighbor Joining [17], which use a matrix representing the evolutionary distances between pairs of species. These methods have the advantage of not requiring much computational effort. However, they are very sensitive to the computed distances and, for large distance values in the matrix (that is, distant species), significant errors are introduced.

1.2. Ant Colony Optimization

Ant Colony Optimization (ACO) is a meta-heuristic proposed by Colorni et al. [2]. ACO is based on the fact that social insects (such as ants, bees and termites) that live in colonies perform specific tasks according to their role in the colony. The self-organization that emerges from the behavior of simple agents (insects) leads the colony, as a whole, to thrive. One of the main tasks that ants need to do is searching for food. Real ants, when searching for food, can find out such resources without visual feedback. Besides, they can adapt themselves to changes in the environment by optimizing the path between the nest and the food source. This fact is the result of stigmergy, that is, a positive feedback, given by the continuous deposit on the path of a chemical substance known as pheromone.

There are many differences between real ants and artificial ants. For instance, artificial ants have memory, they are completely blind and time is discrete [15]. On the other hand, an artificial Ant System allows the simulation of the behavior of real-world ant colonies, such as: artificial ants have preference for trails with large amount of pheromone; shorter paths have a stronger increment in pheromone; there is an indirect communication system between ants (the pheromone trail) that leads them to find the shortest path between the nest and a food source.

1.3. Related work

ACO is a powerful heuristic method that has been applied to many different hard problems, such as combinatorial optimization, data mining and telecommunications routing, among others, but, in the area of Bioinformatics, very few applications have appeared to date.

Korotensky and Gonnet [10] present an alternative method named circular sum, for obtaining the sequence of branches that will give the smallest tree. This method models the problem as a circular Traveling Salesman Problem (cTSP), that is, for a complete tour, the distance from the last city and the first one is added to the tour distance. The tour corresponds to the sequence of species, and the tour distance is the smallest score for this sequence. To construct the tree, a simple idea is used: the correct tree will have the same score found by means of the cTSP. A second algorithm is done to construct trees and compare their scores with the one found by cTSP. This search method is somewhat similar to the maximum parsimony, and, thus, it is computationally expensive for a large number of species.

Kumnorkaew et al. [11] present a new strategy for constructing trees using an ACO algorithm adapted to a Steiner tree problem. In the algorithm, a preprocessing step defines a number of intermediary nodes (ancestral species), by means of the intersection of the input species. From this point on, input species are considered source nodes and the intermediary nodes are considered mandatory passing points. Those authors report that equivalent trees were obtained to those constructed using the Neighbor-joining method. However, it is necessary a strong preprocessing to define proper intermediary points that can be underused.

2. Methodology

2.1. Input data and evolutionary distance

There are several ways to represent input data for phylogenetic tree construction algorithms. Distance-based methods use a square $n \times n$ matrix, where cell (i, j) is the evolutionary distance between species i and j (see example in Fig. 2). Parsimony methods can use two approaches. The first representation is a list of characteristics (attributes) and the corresponding binary information of such characteristics for each species. The second way is based on the multiple alignments of genomic sequences. Finally, probabilistic methods, such as Maximum Likelihood, use as input a sequence of pre-aligned genomic sequences and a model tree.

Considering that distance-based methods tend to be faster than the others, and the distance matrix is similar to a fully-connected graph, this representation was used in this work, as shown in Fig. 2. In this graph, nodes represent the species and edges represent the evolutionary distances between them. This fully-connected graph recalls the simplest instance of the symmetric traveling-salesman problem (TSP). Although simple, this is a hard combinatorial problem. Dorigo and colleagues [2,3] were the first to successfully apply ACO to such problem and this work follows the same principles.

The main drawback of the distance matrix is the need of a metric to compute the evolutionary distance between species. The simplest way to do so is the direct evaluation of corresponding nucleotides of pre-aligned genomic data, using a substitution matrix to compute the sum of base mutations. This method requests the previous alignment of the genomic sequences of species. In the case of quite distant species (or for sequences of different length), the computed distance between species using this method is not reliable. Consequently, a key point of the proposed methodology in this work is the computation of the distance matrix.

Distances between species can be computed using DNA/RNA evolution models, protein evolution models, Brownian motion-based methods or gene frequencies [6]. Recently, Li et al. [13] have proposed an interesting method based on the fact that coding regions of the DNA of close species have strong correlation. Hence, given two DNA sequences (x and y), the distance between them can be calculated using Eq. (2):

$$d(x, y) = 1 - \frac{K(x) - K(x|y)}{K(xy)} \tag{2}$$

where $K(x|y)$ is the conditional Kolmogorov complexity of x given y; $K(x)$ is defined as $K(x|\varepsilon)$, where ε is a null sequence and $K(xy)$ is the Kolmogorov complexity for the concatenation of x and y.

2.2. The basic ACO-based model

As mentioned before, thanks to the convenient representation of a phylogenetic tree as a fully connected graph (based on a distance matrix), the problem is transformed into a TSP. An ACO is then modeled to find a suitable solution for this problem.

In the beginning, ants start in a randomly selected node of the graph. Then, they travel across the structured graph and, at each node, a transition function (Eq. (3)) will determine its direction. This equation represents the probability that the k-th ant, standing at node i, goes to node j in its next step [3].

$$P_k(i, j) = \frac{[\tau(i, j)]^{\alpha} \cdot [d(i, j)]^{-\beta}}{\sum\limits_{u \in J_i^k} \left\{ \ [\tau(i, j)]^{\alpha} \cdot [d(i, j)]^{-\beta} \right\}} \tag{3}$$

In Eq. (3), $P_k(i, j)$ is the probability of transition between nodes i and j; $\tau(i, j)$ is the pheromone trail between nodes i and j; $d(i, j)$ is the evolutionary distance between nodes i and j; J_i^k is the set of nodes connected to node i and already visited by the k-th ant; α and β are arbitrary constants. Two terms can be identified in Eq. (3): one that is based on the evolutionary distance between species i and j, and another based on the accumulated experience, corresponding to the pheromone trail. This trail is represented by a matrix

Species	1	2	3	4
1	0.000	0.199	0.837	0.830
2	0.199	0.000	0.893	0.886
3	0.837	0.893	0.000	0.009
4	0.830	0.886	0.009	0.000

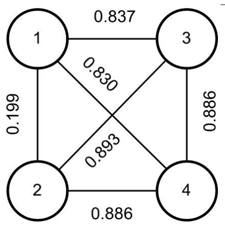

Fig. 2. Example of distance matrix for four hypothetic species (left). Fully connected graph corresponding to the distance matrix (right).

(like that one for the distance between species), whose values are dynamically changed by the algorithm, and determined according to the paths chosen by the ants. Therefore, $\tau(i, j)$ represents the attractiveness of node j, when the ant is at node i.

In the conventional ACO, moves take place between two nodes, and for each step, Eq. (3) is computed. In this work we create an intermediary node n between the current node and that chosen for move. This node will represent the ancestral species of the other two, and it will not be added to the list of species of the tree. The objective of this node is to adjust the distances between the two species (nodes from and to) and the remaining nodes of the graph. More clearly, distance values between species i and j and the remaining species will be recalculated, considering that species i and j were linked by a common ancestor. The new distances between species are recomputed by means of Eq. (4), explained below. The step of joining two nodes with an ancestor species can be done in a preprocessing step. In this case, every ancestor needs to be calculated previously and the distance matrix will increase with ancestor species. Therefore, this preprocessing step and the recalculation of the distance matrix increase the processing time.

$$d_{nu}(i,j) = \begin{cases} d(i,u) + [d(i,u) - d(j,u)].\delta \\ \quad \text{if } d(j,u) > d(i,u) \\ d(j,u) + [d(j,u) - d(i,u)].\delta \\ \quad \text{if } d(i,u) > d(j,u) \end{cases} \qquad (4)$$

In Eq. (4), i and j are two species that are already grouped; n is the ancestral species of i and j, that is, an intermediary node between them; u is the node that represents a species that will be grouped in the current step of the algorithm; $d_{nu}(i,j)$ is the distance between the intermediary node n and node u; $d(i,u)$ is the original distance between nodes i and u; $d(j,u)$ is the original distance between nodes j and u; and δ is a user-defined parameter related to the closeness between a given species and its ancestral.

Figure 3 shows an example of the construction of the intermediary node. In Fig. 3(a), the node n is the common ancestral of nodes 1 and 2, and $\delta = 0.5$, since the distances from n to the other nodes are the same. In Fig. 3(b), the same graph is represented for $\delta < 0.5$, when node n is closer to the source node. In Fig. 3(c), $\delta > 0.5$ and, thus, node n is closer to the target node.

The previous procedure is repeated until all nodes belong to the list of already visited nodes, and then, a path is constructed. The score of this path is given by the sum of the transition probabilities of the adjacent nodes of the path. A cycle of the ACO algorithm is completed when all ants have traversed the graph. At the end of each cycle the pheromone matrix is updated, as explained below.

2.3. Pheromone updating

Paths constructed by the ants are used for updating the pheromone trail. The increment of the pheromone trail is done in all nodes belonging to at least one path, created in an execution cycle. This is an elegant way to avoid fast convergence to a local maximum in the search space. The pheromone trail matrix is updated according to Eq. (5):

$$\tau(i,j) = \rho \cdot \tau(i,j) + (1-\rho) \cdot \Delta\tau(i,j) \qquad (5)$$

where ρ is the pheromone evaporation rate, which reduces the persistence of the environment to the ants. In this work, the pheromone increment rate, $\Delta\tau(i,j)$, was devised to allow an increment proportional to all the obtained paths, given by the division of the current path by the best path found up to the current cycle, as shown in Eq. (6):

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

where $c(t)$ is the path constructed by an ant, up to time t; $S_{c(t)}$ is the score of path $c(t)$ and S_{best} is the score of the best path found up to the current cycle.

For a given path, $S_{c(t)}$ is the quality measure of the solution. This parameter is based on the sum of the transition probabilities of the nodes chosen by the ant

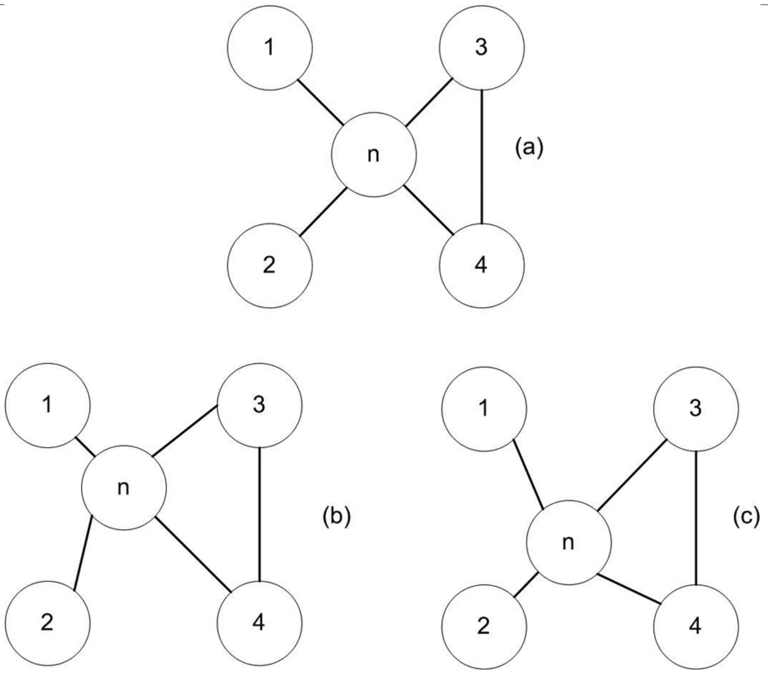

Fig. 3. Example of the construction of the path when an intermediary node is inserted in the original graph: (a) original graph with the inserted node n; (b) node n closer to the origin node; (c) node n closer to the target node.

during the path, and is defined by Eq. (7):

$$S_{c(t)} = \sum_{i=0}^{n}\sum_{j=0}^{n} \begin{cases} P(i,j) & \text{if } i,j \in c(t) \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

By using this procedure, ants traverse the graph and, at the end of the execution of a predefined number of cycles, it is possible to reconstruct the tree using the best path found.

2.4. Reconstruction of the phylogenetic tree

After the execution of the ACO algorithm, as detailed above, we obtain a linear sequence of species (the best path in the graph) and a measure of closeness between them using the pheromone matrix.

The construction of the phylogenetic tree is the next step and starts by finding, for each pair (i, j) of species of the best path, which one has the largest value in the pheromone matrix. This pair is grouped together, then forming a common ancestral species for the pair (i, j). This procedure is repeated until all species have been grouped. The central idea in this procedure is the fact that close species in the graph will be visited by ants more frequently and, consequently, the path between them will accumulate more pheromone. Figure 4 shows a detailed pseudocode of the algorithm for the reconstruction of phylogenetic trees, based on the pheromone matrix M.

3. Computational experiments and results

3.1. Data sets

To evaluate the methodology proposed in this work we used two data sets. The first is a set mitochondrial DNA sequences (mtDNA) from 20 species of mammalians, previously used in another studies (see, for instance, Cao et al. [1]). The test using this data set aimed at evaluating the accuracy of the proposed method, when compared with a consensus tree. Also, we used this data set to find the best set of values for the control parameters of the ACO.

The second data set used was especially constructed for this work. This data set was composed by the complete mitochondrial genomes (mtDNA) of 470 species found in the NCBI site[2] (excluding plasmids). The test using this data set aimed at evaluating the scalability of the proposed ACO algorithm, by observing how the performance of the system behaves as the number of species increases.

[2] http://www.ncbi.nih.gov.

```
WHILE (there are species NOT grouped)
        SEARCH a pair of species (i,j) for which M(i,j) is maximum;
        IF (species i already grouped) THEN
                Substitutes species i by its oldest ancestral y;
        ENDIF;
        IF (species j already gruped) THEN
                Substitutes species j by its oldest ancestral y;
        ENDIF;
        GROUP species i and j;
        SET y as a new species;
        COMPUTE new distance between y and i;
        COMPUTE new distance between y and j;
        CLEAR M(i,j);
ENDWHILE
```

Fig. 4. Pseudocode for the reconstruction of a phylogenetic tree, based on a pheromone matrix M.

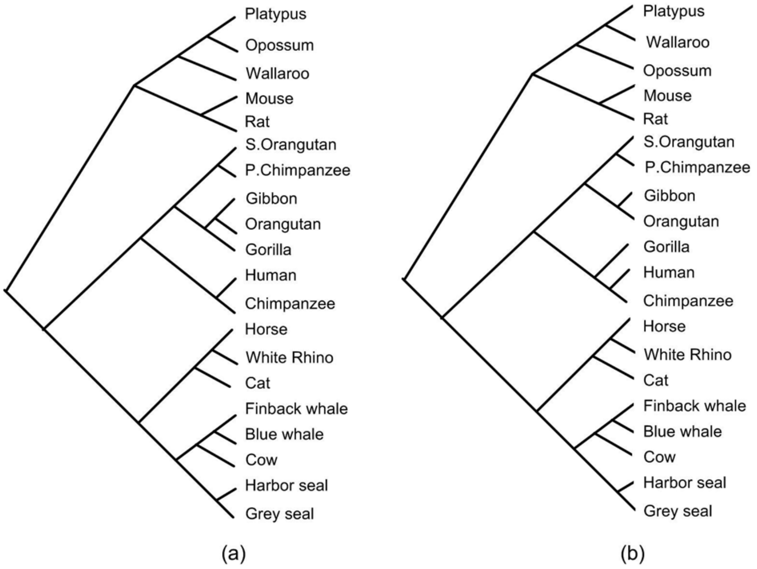

Fig. 5. Comparison between the tree obtained by the proposed method (a) and the consensus tree presented by Cao et al. [1] (b).

3.2. *Sensitivity of the control parameters*

Using the first data set, a number of preliminary experiments was done with different values of the control parameters of the ACO. These experiments were conducted so as to observe the performance of the system as function of the values of the control parameters.

Parameter α controls the exploration of the search space by means of weighting the importance of the pheromone trail in the decision of an ant when it arrives to a branch. Parameter β defines the relative importance of the distance between species in the transitions between nodes. We tested the algorithm with combinations of α and β between 1 and 5. Empirically, we discovered that the algorithm is more sensitive to high values of α, leading to a fast convergence to a local maximum. We also observed that, for better performance, β must be higher than α. However, values too high lead the algorithm to converge to a degenerated tree that groups all species sequentially.

The pheromone trail evaporation is controlled by the parameter ρ, which is influenced by the number of ants (k) and the number of cycles (c). These parameters are directly related to the stigmergy that drives the behavior of ants to find a solution for the problem. We tested ρ between 0.1 and 0.9, in steps of 0.1. Experimentally, we observed that the higher the value of ρ, the smaller the topological distances obtained (with the other parameters fixed), and values lower than 0.2 make the algorithm to find trees with undesired large distances between branches. It is supposed that this is a consequence of the convergence to a local maximum, right in the beginning of the run. For these experiments, k was varied between 50 and 500, in steps of 50; and c was varied between 5 and 50, in steps of 5. Notice that the product $c.k$ reflects the amount of computational effort of the algorithm in a given run.

The evolutionary distance between an ancestral and two descendent species is controlled by parameter δ. This parameter was varied between 0.3 and 0.7, in steps of 0.1. The best tree found was obtained using $\delta = 0.5$, meaning that the distance between the ancestral species and the two descendants is the same for both branches.

For all experiments, we used a reference tree (consensus tree) presented by Cao et al. [1], as the optimal solution. The comparison between a given tree, obtained with a specific set of parameters, and the consensus tree, was done using the Robinson-Foulds method [16]. Although a number of methods for measuring the distance between trees has been proposed [6], the Robinson-Foulds method seems to be one of the most popular. The set of control parameters of our algorithm that obtained the smallest distance to the consensus tree was considered the default. These values are: $\alpha = 1$, $\beta = 2$, $\rho = 0.9$, $\delta = 0.5$, $k = 500$ and $c = 50$.

3.3. Comparison with other methods

To evaluate the results of our approach, we computed the topological distance between the tree generated by our ACO and other methods, namely, Hypercleaning and Fitch [7]. To do so, we used PHYLIP,[3] a widely used package for phylogeny. This computation was done for the first data set, to compare with the consensus tree [1]. Another approach found in the literature [13] used the Neighbor-joining method. Therefore, Table 1 shows the topological distances, calculated using the Robinson-Foulds method [16], for all four approaches (ACO, Fitch, Hypercleaning and Neighbor-joining) regarding to the reference tree.

Table 1
Topological distances, using the Robinson-Foulds method, for the four approaches relative to the consensus tree

Method	Topological distance
ACO	5
Fitch	15
Neighbor-Joining	17
Hypercleaning	4

Our proposed ACO obtained a topology as good as the most robust, although expensive, method (Hypercleaning), and better then the one found by the well-known Fitch method. In Fig. 5 it is shown the comparison of the best tree obtained by our ACO method and the reference tree. There are two small differences between these trees: the order of ancestrals for Platypus, Opossum and Wallaroo, and the branch where Gorilla is located.

3.4. Processing time

The second data set is much larger than the previous one, and it was used to evaluate the processing time of the proposed method as the number of sequences increase. Since this is the first use of this data set, there is no other published work to compare performance with our proposed approach. However, a simple comparison can be done with other methods based on distance matrix.

For this test, we used a desktop PC based on a Celeron 1.1 GHz processor, with 512 MB of RAM. Figure 6 presents the processing time as a function of the number of species to be grouped in a phylogenetic tree by the algorithm. The parameters used in this experiment were those previously defined as default. The same test was repeated using Fitch and Neighbor-joining methods, but for a smaller number of sequences than those of Fig. 6. The growth of the processing time for the three algorithms is shown in Fig. 7. From these two figures, is clear to see that for both, Fitch and Neighbor-joining methods, the processing time tends to grow exponentially, whereas our proposed method tends to grow polynomially. Indeed, a second-order

[3] This software is freely available in the Internet in the site: http://evolution.gs.washington.edu/phylip.html.

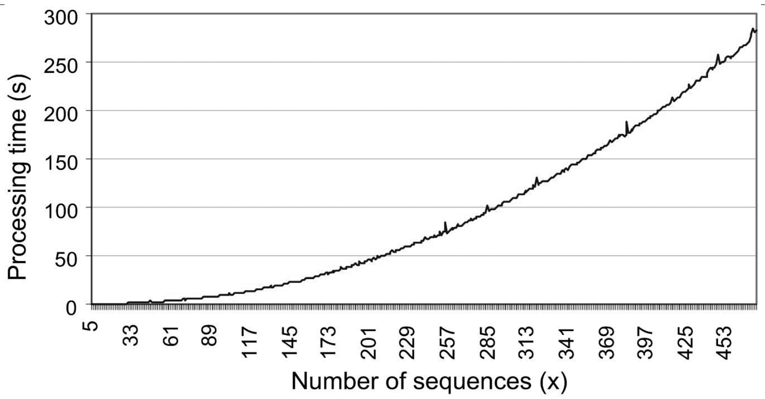

Fig. 6. Processing time of the proposed method as a function of the number of sequences to be grouped in a phylogenetic tree.

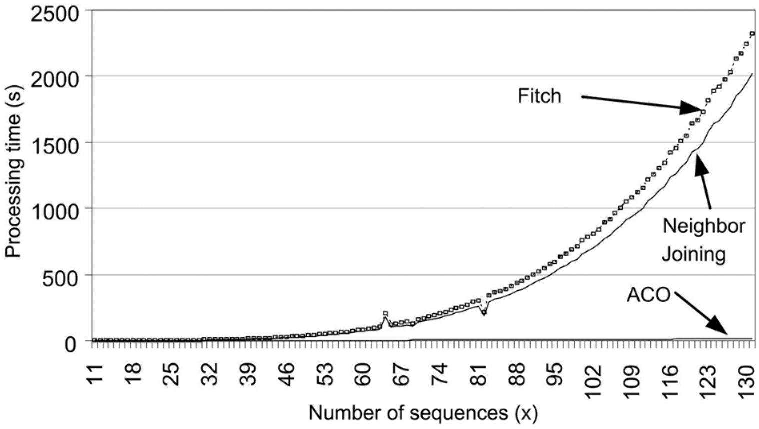

Fig. 7. Comparison of the processing time for the proposed ACO, Fitch and Neighbor-joining methods.

polynomial fitted the data of our approach in Fig. 6, with $R^2 = 0.9997$. Extrapolating the curves of Fig. 7 for a larger number of species, say 1000, our ACO-based approach will need something around 850 seconds of processing time, while the other methods will need more than 54000 seconds.

4. Discussion and conclusions

This work presented a new method for the construction of phylogenetic trees using an Ant Colony Optimization-based approach. We tested our approach with two data sets. For the first data set, the obtained phylogenetic tree was closer to the consensus tree, compared with the trees generated by other established distance-based methods, except Hypercleaning. Hypercleaning obtained a slightly better result than ours. However, it should be noted that Hypercleaning is not a method for phylogenetic tree construction like the other mentioned here. In fact, it is an interactive method that refines trees obtained by any other method, such as Neighbor-joining. Although this strategy works fine for a small number of sequences, it cannot be used for a large number of sequences, due its computational cost.

We examined the sensitivity of the algorithm to changes in their control parameters using the first data set. We observed that the performance of the algorithm is strongly influenced by the value of those parameters. Although we have done many tests so as to propose default values for the control parameters, future work will focus on exhaustive tests, hoping to find optimal values using a broader range of input data. Therefore, it is fair to expect even better performances if an optimized set of parameters can be found.

Using the second data set, the proposed method was also compared with other techniques regarding the processing time. This is an important point of this work since scalability is an important issue in Bioinformatics. Fitch and Neighbor-joining are efficient and fast methods for a small number of sequences, as shown in Fig. 7. However, when a large number of sequences must be analyzed, their processing time may grow prohibitively. Besides, the analysis of Fig. 6 suggests that the embedded complexity of our method tends to be polynomial rather than exponential, at least for the range of sequences tested. Therefore, the proposed method can be an effective alternative for large-scale phylogeny analysis. Overall, results are very promising and encourage further developments.

Besides more tests with other data sets, future work will include the investigation of self-adaptive parameters for the ACO, an emerging technique in evolutionary algorithms [14], and an improved methodology to deal with aligned and non-aligned sequences.

Acknowledgements

Authors would like to thanks the Brazilian National Research Council (CNPq) for the financial support, under grants: 305720/04-0, 475049/03-9 and 402018/03-6.

References

[1] Y. Cao, A. Janke, P.J. Waddell, M. Westerman, O. Takenaka, S. Murata, N. Okada, S. Pääbo and M. Hasegawa, Conflict among individual mitochondrial proteins in resolving the phylogeny of eutherian orders, *Journal of Molecular Evolution* **47** (1998) 307-322.

[2] A. Colorni, M. Dorigo and V. Maniezzo, *Distributed optimization by ant colonies,* in: Proc. of ECAL'91 European Conference on Artificial Life, 1991, 134–142.

[3] M. Dorigo and L.M. Gambardella, Ant colonies for the traveling salesman problem, *Biosystems* **43**(2) (1997), 73–81.

[4] M. Dorigo, E. Bonabeau and G. Theraulaz, *Swarm Intelligence: from Natural to Artificial Systems,* Oxford University Press, 1999.

[5] J. Felsenstein, Maximum likelihood estimation of evolutionary trees from continuous characters, *American Journal of Human Genetics* **25** (1973), 471–492.

[6] J. Felsenstein, *Inferring Phylogenies,* Sinauer Associates, 2004.

[7] W. Fitch and E. Margoliash, The construction of phylogenetic trees, *Science* **155** (1967), 279–284.

[8] G.H. Gonnet, New algorithms for the computation of evolutionary phylogenetic trees, in: *Computational Methods in Genome Research,* S. Suhai, ed., Plenum Press, 1994, pp. 153–161.

[9] J. Kim, Large-scale phylogenies and measuring the performance of phylogenetic estimators, *Systems Biology* **47** (1998), 43–60.

[10] C. Korostensky and G.H. Gonnet, Using traveling salesman problem algorithms for evolutionary tree construction, *Bioinformatics* **16** (2000), 619–627.

[11] M. Kumnorkaew, K. Ku and P. Ruenglertpanyakul, *Application of ant colony optimization to evolutionary tree construction,* in: Proc. of 15^{th} Annual Meeting of the Thai Society for Biotechnology, 2004.

[12] A.R. Lemmon and M.C. Milinkovitch, The metapopulation genetic algorithm: an efficient solution for the problem of large phylogeny estimation, *Proceedings of the National Academy of Sciences* **99** (2002), 10516–10521.

[13] M. Li, J.H. Badger, X. Chen, S. Kwong, P. Kearney and H. Zhang, An information based sequence distance and its application to whole mitochondrial genome phylogeny, *Bioinformatics* **17**(2) (2001), 149–154.

[14] M.H. Maruo, H.S. Lopes and M.R.B.S. Delgado, Self-adapting evolutionary parameters: encoding aspects for combinatorial optimization problems, in: *Evolutionary Computation for Combinatorial Problems,* G.R. Raidl and J. Gottlieb, eds, LNCS 3448, 2005, pp. 154–165.

[15] R.S. Parpinelli, H.S. Lopes and A.A. Freitas, Data mining with an ant colony optimization algorithm, *IEEE Transactions on Evolutionary Computation* **6** (2002), 321–332.

[16] D.F. Robinson and L.R. Foulds, Comparison of phylogenetic trees, *Mathematical Biosciences* **53** (1981), 131–147.

[17] N. Saitou and M. Nei, The neighbor-joining method: a new method for reconstructing phylogenetic trees, *Molecular Biology and Evolution* **4** (1987), 406–425.

[18] L. Salter and D.K. Pearl, Stochastic search strategy for estimation of maximum likelihood phylogenetic trees, *Systematic Biology* **50** (2001), 7–17.

[19] R.R. Sokal and C.D. Michener, A statistical method for evaluating systematic relationships, *University of Kansas Scientific Bulletin* **28** (1958), 1409–1438.