Inferring Phylogenetic Trees Using a Multiobjective Artificial Bee Colony Algorithm

Sergio Santander-Jiménez, Miguel A. Vega-Rodríguez,
Juan A. Gómez-Pulido, and Juan M. Sánchez-Pérez

University of Extremadura,
Department of Technologies of Computers and Communications,
ARCO Research Group,
Escuela Politécnica, Campus Universitario s/n, 10003. Cáceres, Spain
{sesaji,mavega,jangomez,sanperez}@unex.es

Abstract. Phylogenetic Inference is considered as one of the most important research topics in the field of Bioinformatics. A variety of methods based on different optimality measures has been proposed in order to build and evaluate the trees which describe the evolution of species. A major problem that arises with this kind of techniques is the possibility of inferring discordant topologies from a same dataset. Another question to be resolved is how to manage the tree search process. As the space of possible topologies increases exponentially with the number of species in the input dataset, exhaustive methods cannot be applied. In this paper we propose a multiobjective adaptation of a well-known Swarm Intelligence algorithm, the Artificial Bee Colony, to reconstruct phylogenetic trees according to two criteria: maximum parsimony and maximum likelihood. Our approach shows a significant improvement in the quality of the inferred trees compared to other multiobjective proposals.

Keywords: Artificial Bee Colony, Swarm Intelligence, Phylogenetic Inference, Multiobjective Optimization.

1 Introduction

As well as a wide range of topics in Bioinformatics, searching for the best phylogenetic trees which describe the evolution of species is considered as an NP-Hard problem. Several heuristic-based proposals emerged to deal with the computational complexity required by optimality criteria methods like maximum parsimony and maximum likelihood [1]. However, these approaches only consider a single objective to be optimized, so the inference process is carried out in agreement with the chosen criterion. As a result, given a same input dataset, the phylogenies obtained using different methods may be inconsistent with each other. Recent studies have considered the Phylogenetic Inference as an ideal problem to be addressed by multiobjetive optimization techniques [2].

In this paper we try to resolve the Phylogenetic Inference problem according to two metrics: parsimony and likelihood. For this purpose, we define a multiobjective adaptation of the Artificial Bee Colony (ABC), a Swarm Intelligence

M. Giacobini, L. Vanneschi, and W.S. Bush (Eds.): EvoBIO 2012, LNCS 7246, pp. 144–155, 2012.

algorithm based on the colective behaviour of the honey bees [3]. Bioinspired Computing arises as a response to those optimization problems that cannot be solved in reasonable times using classic algorithmic techniques. We have chosen the ABC algorithm because of the promising results which have been reported for a variety of optimization problems [4]. In order to test this algorithm, we have performed a number of experiments on four nucleotide data sets. To illustrate our experimental results, we show the parsimony and likelihood values scored by our Pareto trees and we compared them with the solutions reported by other authors. Additionally, the multiobjective perfomance of our algorithm is evaluated by using the widely used hypervolume metrics.

This paper is organized in the following way. In the next section, we give a brief overview of the published methods for inferring phylogenies. Section 3 explains the basis of Phylogenetic Inference, focusing on the definition of maximum parsimony and maximum likelihood methods. Section 4 describes the proposed algorithm, the Multiobjective Artificial Bee Colony (MOABC). In Section 5 we show, explain and compare the experimental results of our algorithm. Finally, in Section 6 we detail some concluding remarks and define future research lines.

2 Related Work

Phylogenetic Inference has been addressed from different perspectives throughout the years. The first approaches described a number of explicit steps to quickly reconstruct phylogenetic trees from input data. These procedures, known as *algorithmic methods*, do not provide an evaluation function to assess the quality of the generated trees. To overcome this issue, new methods based on optimality measures were proposed. The main goal to achieve with these approaches was the definition of an objective function to be considered in the inference process, generating optimal trees according to some criteria [5]. Some of the most popular *optimality criteria methods* are maximum likelihood and maximum parsimony.

Inferring phylogenetic trees using optimality criteria requires higher processing times than algorithmic methods. Thereby, evolutionary strategies were proposed to deal with this question. The first attempt to apply these ideas was reported by Matsuda in 1995 [6]. Three years later, Lewis developed a genetic algorithm for maximum likelihood inference taking as input nucleotide sequences [7]. This approach laid the foundations for future phylogenetic analyses. Evolutionary algorithms for inferring phylogenetic trees by the maximum parsimony criterion can be also found in the literature [8].

All the previously mentioned approaches suffer from one major drawback: the phylogenetic analyses are performed under a single objective assumption, so the generated topologies may present conflicting ancestral relationships for a same dataset, in agreement with the chosen criterion. Multiobjective optimization emerged as an ideal solution to this [9], allowing new developments to resolve this issue. Coelho et al. published a multiobjective immune-inspired algorithm for inferring phylogenies by minimizing two objective functions: minimal evolution and mean-squared error [10]. In 2007, Cancino and Delbem presented

PhyloMOEA [11], a multiobjective genetic algorithm based on the maximum parsimony and maximum likelihood criteria. The results reported by Cancino and Delbem have motivated this research: a Multiobjective Artificial Bee Colony algorithm for Phylogenetic Inference.

3 Basis of Phylogenetic Inference

Phylogenetic Inference encloses a wide range of estimation techniques that aim to describe ancestral evolutionary relationships among a collection of organisms [5]. These methods take as input a set of n sequences of N characters (often known as sites), which belong to an alphabet α. For example, the alphabet for a DNA-based analysis is $\alpha = \{A, C, G, T\}$, which represents the nucleotide bases Adenine, Cytosine, Guanine and Thymine. These methods use as input molecular characteristics of the organisms under consideration. The output of the inference process is known as *phylogenetic tree.*

A phylogenetic tree is a mathematical structure that represents a hypothesis of the evolution of species. It defines ancestor-descendant relationships among species to explain the data and represents them in a hierarchical tree topology. We can distinguish the following components in a phylogenetic tree:

- Terminal nodes or leaves. They represent the results of the evolutionary history, this is, the input data.
- Internal nodes. They represent hypothetical organisms whose evolution resulted in the species considered as input of the inference process.
- Branches. They indicate an ancestral connection between two nodes. The branches in a tree can be associated to a *branch length* value that usually represents either evolutionary time or molecular changes needed for the evolutionary process.

When a phylogenetic tree has a common ancestor that defines the origin of the phylogeny, we will say that this tree has a *rooted topology.* If this common ancestor does not exist and the direction of natural process cannot be defined, the tree will have an *unrooted topology* [11].

3.1 Methods for Inferring Phylogenies

As we remarked in Section 2, we can find a wide range of methods for inferring phylogenies in the literature. The search for optimal phylogenetic trees is a well-known NP-Hard problem due to the exponential growth of the tree search space in accordance with the number of species. Let n be the number of species to be processed, the number of possible unrooted topologies is given by [12]:

$$\frac{(2n-5)!}{(n-3)!2^{n-3}} \tag{1}$$

This expression means that searching for optimal trees using as input a set of ten or more organisms is cost prohibitive in terms of computational time. The way to address this problem is the development of evolutionary and bioinspired approaches for inferring optimal topologies in acceptable times.

In the following subsections we will introduce the basis of two of the most used criteria-based methods for phylogenetic reconstruction: maximum parsimony and maximum likelihood analysis.

Maximum Parsimony Approach. Cladistic methods based on the parsimony criterion [1] aim to find those phylogenies that minimize the amount of molecular changes needed to explain the observed data. In a maximum parsimony analysis, the simpler the explanation for natural evolution is, the better the parsimony score will be for that phylogenetic tree. This classic approach is inspired by the Occam's razor principle, which affirms that the simplest explanation for an specific phenomenon will be more plausible than other possible hypotheses.

The parsimony score for a phylogenetic tree τ, inferred from a set of n nucleotide sequences characterized by N aligned sites, is given by the following equation [11]:

$$P(\tau) = \sum_{i=1}^{N} \sum_{(a,b)\in B(\tau)} C(a_i, b_i) \tag{2}$$

where (a, b) is a branch in set B which defines an ancestral relationship between the nodes a and b, a_i and b_i the state or value of the ith site on the sequences for a and b, respectively, and $C(a_i, b_i)$ the cost of evolving from the state a_i to b_i. In a maximum parsimony approach, we will prefer those trees which *minimize* this value, because they would represent a simpler explanation to the observed data. In order to compute the parsimony score, we can find a wide range of proposals in the literature. In this work, we will use the algorithm proposed by Fitch [13] to assess the parsimony of a phylogenetic tree.

Maximum Likelihood Approach. In Phylogenetics, the term likelihood refers to an statistical measure that assesses the probability of the observed data given an evolutionary history described by a tree topology. The main goal in a maximum likelihood approach is the reconstruction of that phylogenetic tree which represents the most likely evolutionary history of the species [1]. In a maximum likelihood analysis, we must bear in mind:

1. The topology of the phylogenetic tree.
2. The branch length values.
3. The molecular evolutionary model.

An evolutionary model, also known as substitution model, describes the probabilities of change from a given state to other one on the molecular sequences of two related organisms. Numerous evolutionary models can be found in the literature (such as JC69, F84, HKY85...) [1]. The likelihood value for a phylogenetic tree will be highly related to the chosen substitution model.

We can formulate the likelihood of a phylogenetic tree as follows. Let τ be the phylogenetic tree to be evaluated, D the set of N-site molecular sequences, D_i the ith state value on the sequences and m the substitution model. The likelihood score is the conditional probability of the data given τ and m [1]:

$$L[D, \tau, m] = \Pr[D|\tau, m] = \prod_{i=1}^{N} \prod_{j=1}^{B} (r_i t_j)^{n_{ij}} \tag{3}$$

where B is the set of branches for τ, r_i the probability of change for the state i, t_j the length of the branch j and n_{ij} the number of state changes in the branch j for the character i. The likelihood is an objective to be *maximized*. The higher the likelihood score for a tree is, the more likely the evolutionary hypothesis will be. We can use the Felsenstein algorithm [14] to compute likelihood.

4 Multiobjective Artificial Bee Colony

In the previous section, we have defined the basis of Phylogenetic Inference and performed a quick review of the methods which define the metrics to be used. Now, we will explain the main features of our proposal, a multiobjective adaptation of the Artificial Bee Colony algorithm for inferring phylogenetic trees.

4.1 Artificial Bee Colony Features

The Artificial Bee Colony is a Swarm Intelligence algorithm proposed by D. Karaboga [3] in 2005. He developed a method to resolve classical optimization problems inspired by the collective behaviour of honey bees. Swarm Intelligence algorithms focus on the definition of a collection of individuals who assume a role in the swarm. These individuals perform their activities and interactuate with others to resolve a problem. This behaviour is governed by well-defined rules and allows the swarm to obtain a collective intelligence. The result is the design of new bioinspired algorithms to address a wide range of problems. Recently, the ABC algorithm has been used to resolve several optimization problems, improving the results of classical evolutionary approaches [4]. This algorithm is inspired by the behaviour of three groups of bees in the hive:

- Employed bees. Employed bees aim to look for and exploit food sources. These bees can examinate the neighbourhood of the current food source they are exploiting and find other new sources.
- Onlooker bees. The information gathered by the employed bees about food sources will be used by onlooker bees to select the most promising sources. These interactions among bees take place in the dancing area. Onlooker bees will decide the sources to be exploited in accordance with the quality of them, denoted by the dances performed by employed bees.
- Scout bees. These bees looks randomly at their environment for new undiscovered food sources. The main purpose of these searches is to avoid the absence of food in the hive when the sources found by other bees are exhausted.

Applying the ABC algorithm to optimization problems, we can identify the bees as the individuals in the population, the food sources as possible solutions to the problem, and the nectar they contain as the fitness of these solutions.

4.2 A Multiobjective Artificial Bee Colony Algorithm for Phylogenetic Inference

In this paper, we propose a multiobjective version of the ABC algorithm applied to Phylogenetic Inference. The main goal is to find those phylogenetic trees which represent a consensus between the maximum parsimony and maximum likelihood criteria. From a multiobjective perspective, these solutions cannot be evaluated in a traditional way because they must simultaneously consider conflicting criteria. To resolve this issue, we apply the *dominance* concept [9]: a solution dominates other one if and only if the first solution has better or equal scores in all considered objectives than the second one and, at least, it is better in one of them. Multiobjective metaheuristics try to obtain those non-dominated solutions which are closer to the optimal solutions to the problem, the set of Pareto-optimal solutions. If we represent Pareto solutions in the value space of n objective functions, the resultant n-dimensional curve is known as Pareto front.

The MOABC algorithm takes as input the following parameters:

1. swarmSize. Population size.
2. maxIterations. Iterations of the main loop to be performed.
3. limit. Control parameter defined to avoid population stagnation.
4. mutation. Mutation rate to be applied over the found solutions to generate new ones.

Our proposal will generate as output multiobjective phylogenetic trees according to the parsimony and likelihood metrics, this is, a set of non-dominated Pareto solutions. Algorithm 1 shows the pseudocode for the MOABC.

The MOABC begins with the initialization of employed bees, which represent the first half of the population. For this purpose, random phylogenetic trees are selected from a repository of 1000 trees generated by bootstrap analysis [1] over each dataset. 500 phylogenetic trees are inferred by maximum parsimony analysis using the DNAPARS software from PHYLIP [15]. The remaining 500 trees are generated by maximum likelihood analysis performed with PhyML [16]. After selecting the initial trees, parsimony and likelihood scores are computed by using the Fitch and Felsenstein algorithms. We use the TreeTemplate class from the C++ libraries for bioinformatics BIO++ [17] to encode phylogenetic trees.

We can differentiate three sections in the MOABC loop. Firstly, employed bees search for solutions in the neighbourhood (lines 6-12 of Algorithm 1). For each employed bee, its associated solution is compared to the result of mutating it. Mutation is carried out by applying Nearest Neighbour Interchange (NNI) topological changes [1] (for parsimony treatment) and modifying randomly selected branch lengths using a gamma distribution [7] (for likelihood treatment),

Algorithm 1. MOABC Pseudocode

```
1: /* Initializing the swarmSize/2 employed bees */
2: C ← initializeAndEvaluatePopulation(swarmSize/2)
3: ParetoFront ← 0
4: i ← 0
5: while i < maxIterations do
6:    for j = 1 to swarmSize/2 do
7:       /* Employed bees: searching for solutions in the neighbourhood*/
8:       newEmployedBee ← generateNeighbour(C[j],mutation)
9:       if MOFitness(newEmployedBee) < MOFitness(C[j]) then
10:         C[j] ← newEmployedBee
11:      end if
12:   end for
13:   /* Generating the probability vector */
14:   probVector ← calculateSelectionProbabilities(C)
15:   /* Generating onlooker bees according to the probability vector */
16:   for j = (swarmSize/2)+1 to swarmSize do
17:      selectedEmployedBee ← selectEmployedBee(probVector,C)
18:      newOnlookerBee ← generateNeighbour(selectedEmployedBee,mutation)
19:      if MOFitness(newOnlookerBee) ≤ MOFitness(selectedEmployedBee) then
20:         C[j] ← newOnlookerBee
21:      else
22:         C[j] ← selectedEmployedBee
23:      end if
24:   end for
25:   /* Generating scout bees */
26:   for j = 1 to swarmSize/2 do
27:      if C[j].iterations > limit then
28:         C[j] ← generateScoutBee()
29:      end if
30:   end for
31:   /* Sorting the current solutions */
32:   C ← FastNonDominatedSort(C)
33:   /* Saving Pareto solutions */
34:   ParetoFront ← saveSolutions(C, ParetoFront)
35:   i ← i + 1
36: end while
```

both according to the *mutation* rate parameter. The NNI operator takes an internal branch of the tree and executes a swap between the nodes in the subtrees situated at the sides of the chosen branch to generate new topologies. In order to improve the likelihood score, we also apply a gradient descent algorithm to optimize tree branch lengths [18].

Once we have generated the neighbour solution, we must decide which one is better in a multiobjective context. For this purpose, Equation 4 is calculated for each competing solution. MOFitness assigns a score to a solution b according to the number of solutions in the population dominated by b and the solutions that dominates b ([19] includes a more detailed explanation). The tree that minimizes this expression will be the solution assigned to the employed bee.

$$MOFitness(b) = Dominates(b) + isDominated(b) * swarmSize \tag{4}$$

Secondly, onlooker bees (the second half of the population) will decide which solutions must be exploited in accordance with the information provided by employed bees (lines 13-24). For this purpose, current solutions are ordered using two operators taken from NSGAII: *fast non dominated sort* (FNDS) and *crowding distance* [20]. After that, we compute a vector to define selection probabilities for each solution. The better the solution quality is, the higher its selection probability will be. Onlooker bees will verify this vector and choose one of the current solutions. Neighbour trees are computed by applying mutation and compete with the selected ones using MOFitness. Unlike the previous step, a neighbour solution will be saved if it scores a lower or *equal* MOFitness value with regard to the original solution. Allowing equal scores helps to promote population diversity.

Thirdly, scout bees are generated in the next section (lines 25-30). The *limit* parameter plays a key role in this step. If the solution associated to a bee is not improved in *limit* iterations, it must be discarded (local optimum). This individual becomes a scout bee, which will explore the search space for new solutions. Scout bees randomly select phylogenetic trees from the bootstrap repository and improve them by applying deeper NNI moves and branch length optimization. This strategy allows to avoid local optimal by using different starter trees to explore undiscovered regions of the tree search space.

Once these three sections have been completed, the $swarmSize/2$ best phylogenetic trees found in this iteration are assigned to the employed bees as new starter trees and the MOABC loop begins again. The Pareto set is updated with the best non-dominated solutions and, after *maxIterations*, it will contain those trees whose parsimony and likelihood scores are closer to the optimal values.

5 Experimental Methodology and Results

In this section we explain the methodology to configure our algorithm and show experimental results. Parameters values were assigned in agreement with other authors' proposals [11] with which we will compare our results. For the limit parameter, we performed several experiments to decide its optimal value. For each considered limit value (5, 10, 15, 20 and 25), ten independent runs were carried out and the Pareto sets were evaluated using the hypervolume indicator, a multiobjective metrics that indicates the search space area dominated by our Pareto solutions. The results indicated that the best mean hypervolume values were achieved by using the limit=15 value. Table 1 shows MOABC's configuration.

Our approach was tested on four public nucleotide data sets [11]: $rbcL\_55$ contains 55 sequences (1314 nucleotides per sequence) of the rbcL gene from different species of green plants. $mtDNA\_186$ has 186 sequences (16608 nucleotides per sequence) of human mitochondrial DNA. $RDPII\_218$ is composed of 218 sequences (4182 nucleotides per sequence) of prokaryotic RNA. And $ZILLA\_500$ contains 500 sequences (759 nucleotides per sequence) from rbcL plastid gene.

To prove the statistical relevance of our approach, we have conducted a set of experiments consisting on ten executions per dataset. At the end of them, Pareto fronts were evaluated using the hypervolume metrics and the Shimodaira-Hasewaga (SH) test [21]. Meanwhile hypervolume evaluates solutions from a multiobjective perspective, the SH test decides which percentage of these solutions are not significantly worse than optimal phylogenetic trees found by single-objective approaches [11]. Table 2 resumes our experimental results. For each dataset, Figure 1 shows the Pareto fronts which score the hypervolume value closer to the mean hypervolume obtained by the overall experiments.

Table 1. MOABC input parameters

Parameter	Value
maxIterations	100
swarmSize	100
mutation	5%
limit	15
Topological operator	NNI
Substitution model	HKY85

Table 2. Experimental results

Dataset	Pareto	Most Parsimonous		Most Likely		SH Test	
	Solutions	Tree		Tree		Parsimony	Likelihood
		Parsimony	Likelihood	Parsimony	Likelihood		
*rbcL*_55	6	4874	-23969.111	4881	-23961.487	100%	100%
*mtDNA*_186	14	2431	-40319.181	2448	-40240.283	80%	85%
*RDPII*_218	34	41488	-149938.023	42621	-144087.843	8%	26%
*ZILLA*_500	36	16218	-82684.057	16272	-82497.441	98%	61%

Table 3. Hypervolume metrics

Dataset	Minimal		Maximum		Hypervolume	
	Reference Point		Reference Point		Mean (%)	Std. deviation
	Parsimony	Likelihood	Parsimony	Likelihood		
*rbcL*_55	4774	-23495.5	5279	-25941.6	64.87	0.0016
*mtDNA*_186	2376	-39376.4	2656	-43567.2	64.50	0.0005
*RDPII*_218	40658	-140667.6	45841	-162933.1	70.15	0.0193
*ZILLA*_500	15893	-80850.9	17588	-89319.8	65.05	0.0007

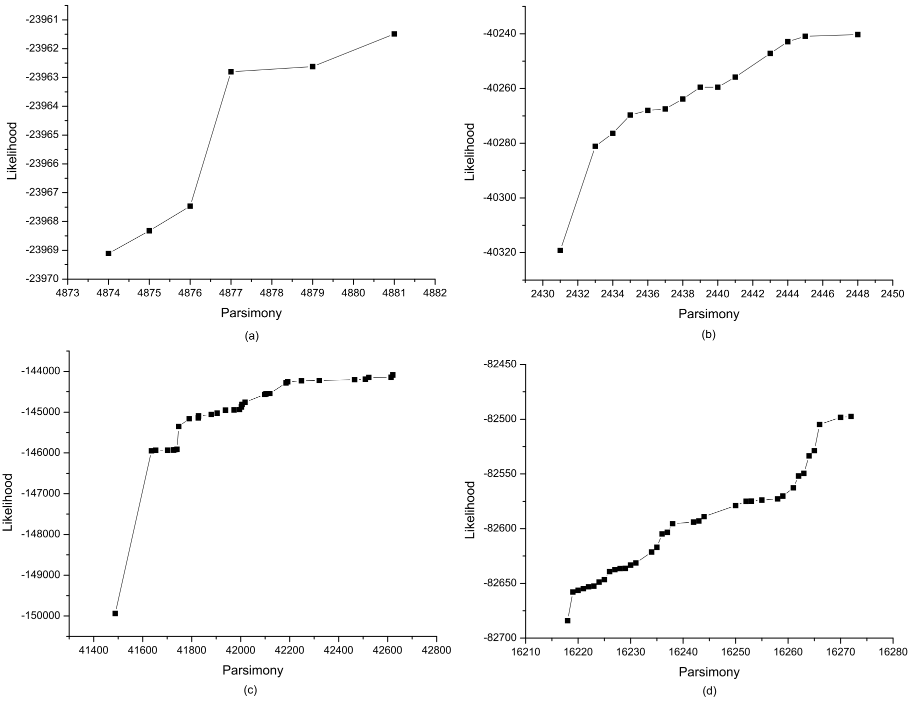

Fig. 1. Pareto fronts for *rbcL*_55(a), *mtDNA*_186(b), *RDPII*_218(c) and *ZILLA*_500(d)

Considering the results for *rbcL*_55, *mtDNA*_186 and *ZILLA*_500, the SH test shows a high acceptance ratio because the inferred phylogenetic trees are in the optimal parsimony-likelihood range. For the *RDPII*_218 dataset, extreme points on Pareto Front define a higher range of possible non-dominated solutions. This fact motivates a lower acceptance rate. Analyzing these results, we can conclude that the SH test denotes that our extreme solutions are relevant from a single-objective perspective. To evaluate the multiobjective performance of our algorithm, we used the well-known hypervolume indicator. Table 3 defines the reference points and shows mean hypervolume values for each dataset. According to Column 6, our solutions cover over 64% of the space defined by the reference points for *rbcL*_55, *mtDNA*_186 and *ZILLA*_500, and a 70.15% for *RDPII*_218. We would like to remark that these results are significantly interesting because they would allow researchers to make future comparisons with other bioinspired multiobjective approaches to Phylogenetic Inference.

5.1 Comparisons with Other Authors

In this subsection we present a comparison between our MOABC and PhyloMOEA, a multiobjective algorithm for Phylogenetic Inference published by Cancino and Delbem [11]. Table 4 shows the parsimony and likelihood scores for the most parsimonious and most likely trees (columns 2-3 and 4-5, respectively) found by the two algorithms for each dataset. Our MOABC improves the results reported by Cancino and Delbem in all datasets. For the *rbcL*_55 instance, the most parsimonious tree found by the two algorithms scores the same parsimony but MOABC improves the likelihood value. This fact demonstrates that our solutions dominate the trees generated by using PhyloMOEA.

In a recent study, Cancino and Delbem suggested the use of a parametric evolutionary model called $HKY85 + \Gamma$ to improve likelihood scores, without changing parsimony values [22]. We have implemented that model in our proposal and carried out new experiments. In Table 5 we can find the likelihood values for the most likely trees found by the two algorithms. Once again, our proposal improves PhyloMOEA's results. Consequently, we conclude that MOABC shows a significant improvement in the quality of the inferred trees.

Table 4. MOABC - PhyloMOEA Comparison

	MOABC			
	Best parsimony		Best likelihood	
Dataset	Parsimony	Likelihood	Parsimony	Likelihood
*rbcL*_55	4874	**-23969.111**	**4881**	**-23961.487**
*mtDNA*_186	**2431**	**-40319.181**	**2448**	**-40240.283**
*RDPII*_218	**41488**	**-149938.023**	**42621**	**-144087.843**
*ZILLA*_500	**16218**	**-82684.057**	**16272**	**-82497.441**
	PhyloMOEA			
	Best parsimony		Best likelihood	
Dataset	Parsimony	Likelihood	Parsimony	Likelihood
*rbcL*_55	4874	-24626.243	4884	-24583.330
*mtDNA*_186	2438	-41004.302	2450	-40894.343
*RDPII*_218	41534	-158724.280	42631	-156595.822
*ZILLA*_500	16219	-87275.281	16276	-86993.825

Table 5. Best likelihood scores using $HKY85 + \Gamma$

Dataset	MOABC	PhyloMOEA
*rbcL*_55	**-21821.480**	-21889.844
*mtDNA*_186	**-39890.140**	-39896.441
*RDPII*_218	**-134149.328**	-134696.535
*ZILLA*_500	**-80965.400**	-81018.060

6 Conclusions

We have studied in this paper a multiobjective adaptation of a Swarm Intelligence algorithm, the Artificial Bee Colony, to infer phylogenetic trees according to the maximum parsimony and maximum likelihood principles. Our approach has been tested on four public nucleotide data sets and a variety of experiments has been carried out. We have evaluated the multiobjective performance of the proposal by computing the hypervolume metrics and reference points have been defined for future comparisons. Experimental results have proved the relevance of our approach, inferring phylogenetic trees which considerably improve the results reported by other authors in the literature. Therefore, we can suggest that multiobjective Swarm Intelligence algorithms offer multiple possibilities to define improved heuristic approaches to Phylogenetic Inference.

As future research lines, we will address the question of how to boost performance by applying Parallel Computing. Bioinspired techniques and Parallelism will allow us to develop heuristic-based algorithms to infer multiobjective phylogenetic trees minimizing execution times. From this perspective, MPI and OpenMP libraries can help the researcher to exploit the characteristics of modern parallel architectures to improve efficiency. Additional future work could be the study of different topological operators (such as Subtree Pruning and Regrafting (SPR) and Tree Bisection and Reconnection (TBR)) and evolutionary models (JC69, F84...) to optimize parsimony and likelihood scores.

Acknowledgements. This work was partially funded by the Spanish Ministry of Science and Innovation and ERDF (the European Regional Development Fund), under the contract TIN2008-06491-C04-04 (the M* project). We would like to thank the Fundación Valhondo Calaff for the financial support offered to Sergio Santander-Jiménez.

References

1. Felsenstein, J.: Inferring phylogenies. Sinauer Associates, Sunderland (2004); ISBN: 0-87893-177-5
2. Handl, J., Kell, D., Knowles, J.: Multiobjective Optimization in Computational Biology and Bioinformatics. IEEE Transactions on Computational Biology and Bioinformatics 4(2), 289–292 (2006)
3. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
4. Karaboga, D., Basturk, B.: A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization 39(3), 459–471 (2007)
5. Swofford, D., Olsen, G., Waddell, P., Hillis, D.: Phylogenetic Inference. Molecular Systematics, vol. 2, pp. 407–514. Sinauer Associates, Sunderland (1996)
6. Matsuda, H.: Construction of phylogenetic trees from amino acid sequences using a genetic algorithm. In: Proceedings of Genome Informatics Workshop, pp. 19–28. Universal Academy Press (1995)

7. Lewis, P.O.: A Genetic Algorithm for Maximum-Likelihood Phylogeny Inference Using Nucleotide Sequence Data. Molecular Biology and Evolution 15(3), 277–283 (1998)
8. Congdon, C.: GAPHYL: An evolutionary algorithms approach for the study of natural evolution. In: Genetic and Evolutionary Computation Conference, pp. 1057–1064 (2002)
9. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester (2001); ISBN: 978-0-471-87339-6
10. Coelho, G.P., da Silva, A.E.A., Von Zuben, F.J.: Evolving Phylogenetic Trees: A Multiobjective Approach. In: Sagot, M.-F., Walter, M.E.M.T. (eds.) BSB 2007. LNCS (LNBI), vol. 4643, pp. 113–125. Springer, Heidelberg (2007)
11. Cancino, W., Delbem, A.C.B.: A Multi-objective Evolutionary Approach for Phylogenetic Inference. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 428–442. Springer, Heidelberg (2007)
12. Poladian, L., Jermiin, L.: Multi-Objective Evolutionary Algorithms and Phylogenetic Inference with Multiple Data Sets. Soft Computing 10(4), 359–368 (2006)
13. Fitch, W.: Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. Systematic Zoology 20(4), 406–416 (1972)
14. Felsenstein, J.: Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach. Journal of Molecular Evolution 17, 368–376 (1981)
15. Felsenstein, J.: PHYLIP (Phylogeny Inference Package) (2000), `http://evolution.genetics.washington.edu/phylip.html`
16. Guindon, S., Dufayard, J.F., Lefort, V., Anisimova, M., Hordijk, W., Gascuel, O.: New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. Systematic Biology 59(3), 307–321 (2010)
17. Dutheil, J., Gaillard, S., Bazin, E., Glémin, S., Ranwez, V., Galtier, N., Belkhir, K.: Bio++: a set of C++ libraries for sequence analysis, phylogenetics, molecular evolution and population genetics. BMC Bioinformatics 7, 188 (2006)
18. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes in C, The Art of Scientific Computing. Cambdrige University Press (1992); ISBN: 0–521–43108–5
19. Weicker, N., Szabo, G., Weicker, K., Widmayer, P.: Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment. IEEE Transactions on Evolutionary Computation 7(2), 189–203 (2003)
20. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
21. Shimodaira, H., Hasegawa, M.: Multiple comparisons of log-likelihoods with applications to phylogenetic inference. Molecular Biology and Evolution 16, 1114–1116 (1999)
22. Cancino, W., Delbem, A.C.B.: A Multi-Criterion Evolutionary Approach Applied to Phylogenetic Reconstruction. In: Korosec, P. (ed.) New Achievements in Evolutionary Computation, pp. 135–156, InTech (2010); ISBN: 978-953-307-053-7