Multi-Criterion Phylogenetic Inference using Evolutionary Algorithms

Waldo Cancino and A.C.B. Delbem
Institute of Mathematics and Computer Science
University of Sao Paulo
Av. Trabalhador Sao-carlense, 400 - Centro
Sao Carlos, SP. Brazil 13560–970
Email: {wcancino, acbd}@icmc.usp.br

Abstract—Various phylogenetic reconstruction methods have been proposed in order to determine the most accurate tree that represents evolutionary relationships among species. Each method defines a criterion for evaluation of possible solutions. This criterion leads the search to the best phylogenetic tree. However, different criteria may lead to distinct phylogenies, which often conflict with each other. In this context, a multi-objective approach can be useful since it could produce a set of optimal trees (Pareto front) according to multiple criteria. We propose a multi-objective evolutionary algorithm, called PhyloMOEA, which is focused on maximum parsimony and maximum likelihood criteria. In experiments, several PhyloMOEA trials were performed using four datasets of nucleotide sequences. For each dataset, the proposed algorithm found a Pareto front representing a trade-off between the criteria used. Moreover, SH-test showed that a number of solutions from PhyloMOEA are not significantly worse than solutions found by phylogenetic programs using one criterion.

I. INTRODUCTION

Phylogenetic inference, which searches for the best tree that explains the evolutionary events from a given dataset, is one of the main problems in computational biology. It is often modeled as a single objective optimization problem using one criterion to evaluate possible solutions. There are several phylogenetic reconstruction methods which use distinct criteria. Some research [1]–[3] has shown important differences in the results obtained by applying distinct reconstruction methods to the same input data. Rokas et al [4] pointed out that there are several sources of incongruity in phylogenetic analysis: the optimality criterion used, data used and evolutionary assumptions concerning data.

A multi-objective approach, which can search for phylogenies using more than one criterion, can be a relevant contribution since it produces solutions which are consistent with all employed criteria. Recently, Handl et al [5] discussed applications of multi-objective optimization in bioinformatics and computational biology problems. Moreover, Poladian and Jermiin [6] suggested that multi-objective optimization can be used in phylogenetic inference from various conflicting input data. The authors showed that this approach reveals sources of such conflicts and provide useful information for a robust inference.

We propose a multi-objective approach for phylogenetic inference using maximum parsimony [7] and maximum likelihood [8] criteria. The algorithm developed to solve such a problem, called PhyloMOEA, is a multi-objective evolutionary algorithm (MOEA) based on the NSGA-II model proposed by Deb et al [9]. The output from PhyloMOEA is a solution set representing a trade-off between the criteria considered.

This paper is organized as follows. Section II provides relevant background information about phylogenetic inference and two of the most popular phylogenetic reconstruction methods: maximum parsimony and maximum likelihood. Section III presents main concepts of Genetic Algorithms (GAs) and their application to phylogeny. Section IV discusses multi-objective optimization problems and shows how GAs can contribute to solve this kind of problems. Section V presents the PhyloMOEA algorithm. Section VI describes the experiments involving four nucleotide datasets and discusses the main results. Finally, Section VII presents conclusions and proposes future work.

II. PHYLOGENETIC INFERENCE PROBLEM

Phylogenetic analysis investigates evolutionary relationships among species. Data used in this analysis usually come from sequence data (nucleotide or aminoacid sequences), although other types of data can be used [10]. Evolutionary relationships are often represented as a leaf-labelled tree, called a phylogenetic tree. In this tree, external nodes refer to actual species in data, internal nodes refer to hypothetical ancestors and branches represent relations among species. Since data used in phylogenetic analysis are obtained from contemporary species, a phylogenetic tree is a hypothesis (of many possible ones) of the evolutionary events in the history of species.

A phylogenetic tree can be rooted or unrooted. In a rooted tree, there is a special node called root that defines the direction of the evolution, determining ancestral relationships among nodes. An unrooted tree shows only the relative positions of nodes without an evolutionary direction. Additionally, tree branches may have an associated length showing genetic distances between connected nodes. Figs. 1 and 2 show a rooted and unrooted tree, respectively.

The main goal of the phylogenetic inference is the determination of a tree that best explains the evolutionary events...
of the species under analysis. Swofford et al. [11] classify phylogenetic reconstruction methods into two categories: algorithmic and optimality criterion methods. The former follows a sequence of well-defined steps to generate a tree without examining many alternatives in the search space. Moreover, algorithmic methods go directly to the final answer quickly producing a tree. Clustering approaches like Neighbor Joining [12] are in this category.

On the other hand, optimization criterion methods incorporate an optimality criterion and a search mechanism. The former defines an objective function that scores every possible solution. The search mechanism examines the tree search space in order to determine the best scored tree according to the criterion used. There are exact and exhaustive search strategies that can find the best scored tree. However, these strategies are only applicable in small datasets due to the tree search space, which increases exponentially with the number of species analyzed. For moderate and large datasets, only heuristic search strategies are feasible but there is no guarantee that the solution found is optimal. Optimality criterion methods usually produce better results [1], although they are slower than algorithmic methods. Examples of optimality criterion methods are maximum parsimony [7], maximum likelihood [8] and least squares [13].

The following sections present a brief review of maximum parsimony and maximum likelihood since they are used in PhyloMOEA.

A. Maximum Parsimony

The parsimony principle states that the simplest hypothesis concerning an observed phenomenon must always be preferred. In phylogenetic inference, parsimony methods search for a tree topology that requires the minimum number of character state changes. This tree, which is the simplest explanation for a given data set, is called a maximum parsimony tree [10].

Let D be a dataset containing n species. Each specie has N sites, where d_{ij} is the character state of specie i at site j. Given a tree T with a node set V(T) and a branch set E(T), the parsimony score of T is defined as:

$$PS(T) = \sum_{j=1}^{N} ps_j,$$

where $ps_j$ is the number of character changes along branches in T for site j. This quantity can be formulated as:

$$ps_j(T) = \sum_{(v,u) \in E(T)} C(v_j, u_j),$$

where $v_j$ and $u_j$ are the character states of nodes v and u at site j for each branch $(u, v)$ in T, C is the cost matrix such that $C(v_j, u_j)$ is the cost of changing from state $v_j$ to state $u_j$. The leaves of T are labelled by character states of species from D, i.e. a leaf representing k-th species has a character state $d_{kj}$ for position j. The following properties can be noted from (1) and (2):

1) Parsimony criterion assumes independence of sites, i.e. each site is evaluated separately;
2) The calculation of the parsimony score only takes into account the tree topology. Thus, the parsimony criterion does not incorporate other information like branch lengths.

It is necessary to determine the character states of internal nodes from T so that PS is minimized. This is the called small parsimony problem, which can be solved by the Sankoff algorithm [14] for an arbitrary cost matrix C. When C satisfies $C_{xy} = 1$ if $x \neq y$ and $C_{xy} = 0$ if $x = y$, the Fitch algorithm [7] can be used.

The task of finding the maximum parsimony tree in the search space is called the large parsimony problem. This problem was proved to be NP-hard [10], however several heuristic techniques have been proposed in order to overcome such a difficulty [15].

B. Maximum Likelihood

Likelihood is a widely-used statistical measurement. It evaluates a probability that a hypothesis could give rise to the observed data [11]. The likelihood of a phylogenetic tree, denoted by $L = P(D|T, M)$, is the conditional probability of the sequence data D given a tree T and a stochastic evolution model denoted by M. Two assumptions are necessary to compute likelihoods:

1) Evolution at different sites is independent;
2) Evolution from different tree lineages is independent, i.e. each subtree evolves separately.

Let $D_i$ be the sequence data set D at site i. $L_i$ is calculated from the product of partial likelihoods from all sites:

$$L = \prod_{i=1}^{N} L_i,$$

where $L_i = P(D_i|T, M)$ is the likelihood at site i. An efficient method to calculate L was proposed by Felsenstein [8] using a dynamic programming approach, where L is obtained by a post-order traversal of T. Usually, it is convenient to work with logarithmic values of L, then (3) results in:

$$\ln L = \sum_{i=1}^{n} L_i$$


In order to maximize $L$ for a given tree $T$, it is necessary to optimize the branch lengths of $T$ and the parameters of $M$. This can be achieved using classical optimization methods [10]. Finding the maximum likelihood tree in the search landscape is a more difficult problem. Moreover, only heuristic approaches [16]–[19] are feasible for moderate or large datasets.

Genetic Algorithms (GAs) are heuristics that can be used in phylogenetic inference. The next section discusses GAs and their application in phylogenetic analysis.

III. GENETIC ALGORITHMS IN PHYLOGENETIC INFERENCE

Genetic Algorithms are search and machine learning techniques inspired by natural selection principles [20]. They have been applied to a wide range of problems in science and engineering [21]. A GA uses a set of individuals, called population, which represents feasible solutions for a given optimization problem. Each individual has an associated fitness value, which is based on the objective function. Individuals use an internal encoding that must be able to store all relevant problem variables and codify all feasible solutions.

First, a GA creates an initial population and calculates the fitness of its individuals. Then, a new population is generated using three genetic operators: selection, crossover and mutation [20]. The selection operator uses individuals' fitness to choose the best candidates to generate the next population. Features of the selected solutions are combined by the crossover operator and new offspring solutions are created. Then, small modifications are performed by the mutation operator at a very low rate. While crossover is useful to explore the search space, mutation can escape from local optima. The average fitness of the new population is expected to improve. This process is repeated until a stop criterion has been reached. Thus, the fitness landscape in combination with the selection process is repeated until a stop criterion has been reached. The solutions found by the GA are in the final population.

Various papers have described the application of GAs to the phylogeny problem focused on one optimality criterion. In general, these studies use maximum likelihood [18], [19], [22], parsimony [23], [24] or distance-based [25] criterion. Experimental results have shown that GAs have better performance and accuracy when compared to heuristics implemented in widely-used phylogenetic software, like PHYLIP [26] and PAUP* [27]. Moreover, GAs are also suitable for use several optimality criteria in order to solve multi-objective optimization problems (MOOP). The following section briefly describes MOOPs and the application of GAs to these problems.

IV. MULTI-OBJECTIVE OPTIMIZATION

A MOOP deals with two or more objective functions that must be simultaneously optimized. In this context, the Pareto dominance concept is commonly used to compare two solutions. A solution $x$ dominates a solution $y$ if $x$ is not worse than $y$ in all objectives and if it is better for at least one. Solving a MOOP implies calculating the Pareto optimal set whose elements, called Pareto optimal solutions, represent a trade-off among objective functions. Pareto optimal solutions are not dominated by any other in the search space. The curve formed by plotting these solutions in the objective function space is entitled Pareto front. If there is no additional information regarding the relevance of the objectives, all Pareto optimal solutions have the same importance. Deb [21] pointed out two fundamental goals in MOOP:

1) To find a set of solutions as close as possible to the Pareto optimal front;
2) To find a set of solutions as diverse as possible.

Many classical optimization techniques have been proposed to deal with MOOPs [21]. The simplest approach transforms a MOOP into a single optimization problem using a weighted sum of objective functions. This strategy only finds a single point in the Pareto front for each weight combination. Thus, several runs using different weight values are required to obtain a reasonable number of Pareto optimal solutions. Nevertheless, this method does not guarantee solution diversity in the frontier. There are other methods to deal with MOOPs, but all of them have limitations, i.e. they need a priori knowledge of the problem, for example, target values; which are not always available.

On the other hand, evolutionary algorithms for multi-objective optimization (MOEAs) have been successfully applied to both theoretical and practical MOOPs [21]. In general, the most elaborated MOEA models are capable of finding a distributed Pareto optimal set in a single run.

The following sections describe PhyloMOEA, the proposed MOEA to solve the phylogenetic inference problem using maximum parsimony and maximum likelihood criteria.

V. A MULTI-OBJECTIVE APPROACH TO PHYLOGENETIC INFERENCE

As mentioned in Section II, the use of various phylogenetic reconstruction methods can produce different results for the same input data. Huelsenbeck [1] performed a study of the main phylogenetic approaches. In this study, most methods performed successfully for simulated datasets generated for four species. However, under some conditions, methods failed to find the true tree producing different answers. Other studies [2], [3], [28] using simulated and real datasets confirmed these results. Consequently, the selection of the reconstruction method is a crucial step in phylogenetic analysis.

Optimality criterion methods are based on only one criterion in order to evaluate possible solutions. Thus, the phylogenetic reconstruction problem is often solved as a single objective optimization problem. As results obtained from diverse phylogenetic methods often disagree, a multi-objective approach, that takes into account several criteria simultaneously, is a feasible alternative. This approach not only allows to determine the best solution according to each criterion separately, but finds intermediate solutions representing a trade-off among criteria used.
This paper formulates the phylogenetic inference problem as a MOOP with two optimality criteria: maximum parsimony and maximum likelihood. In order to solve such a problem, we have proposed a MOEA algorithm called PhyloMOEA, which is based on the NSGA-II [9] model. The main aim of PhyloMOEA is to determine a set of non-dominated solutions (trees), which represents a trade-off between parsimony and likelihood scores. The following subsections describe the proposed algorithm in more details.

A. Internal Encoding

Phylogenetic trees are usually represented using graph data structures [29]. PhyloMOEA uses the Graph Template Library (GTL) [30] to work with unrooted trees. GTL makes easy implementation of genetic operators possible and facilitates the storage of additional information, such as branch lengths. Furthermore, parsimony and likelihood criteria can operate on rooted or unrooted trees.

B. Initial Solutions

PhyloMOEA uses two populations, a parent population and an offspring population, in the same way as NSGA-II. The parent population is denoted as $P_i$, where $i$ refers to the $i$th generation. In the first generation, solutions from $P_1$ are created at random, and, in subsequent generations, $P_i$ stores the best solutions found in the previous $i-1$ iterations. Solutions in $P_i$ are also used to create the offspring population, denoted by $Q_i$, by applying selection, crossover and mutation.

PhyloMOEA can generate initial random trees in $P_1$; however, these trees are often far from maximum parsimony and likelihood trees. In order to overcome this drawback, additional solutions, provided by maximum likelihood, parsimony or bootstrap analysis, can be included in the initial population. This strategy is usually used in GA-based phylogenetic programs [19], [22].

C. Fitness Evaluation

PhyloMOEA evaluates trees in $P_i$ and $Q_i$, using the parsimony and likelihood criteria. The parsimony and likelihood scores are calculated using Fitch [7] and Felsenstein [8] algorithms, respectively. The tree fitness is obtained using two values: a rank and a crowding distance [21].

The rank value is calculated using a non-dominated sorting algorithm [21] applied to $R = P_i \cup Q_i$. This algorithm divides $R$ into several frontiers, denoted by $F_1, F_2, \ldots, F_j$. The first frontier ($F_1$) is formed by non-dominated solutions from $R$. Solutions in $F_1$ are removed from $R$ and the remaining solutions are employed to calculate the next set of non-dominated solutions, denoted by $F_2$. This process is repeated in order to find $F_3$, and so on, until $R$ is empty. The rank value of an individual is the index of the frontier it belongs to.

The crowding distance is useful to maintain the population diversity. It reflects the density of solutions around its neighborhood. This value is calculated from a perimeter defined by the nearest neighbors in each objective.

PhyloMOEA uses a tournament selection which picks two individuals at random and chooses the best one, which has the lowest rank. If both solutions have the same rank, the solution with the longest crowding distance is preferred.

D. Crossover Operator

The crossover operator implemented in PhyloMOEA is the same as [18]. It combines a subtree from two parent trees and creates two new offspring trees. Given trees $T_1$ and $T_2$, this operator performs the following steps:

1) Prune a subtree $s$ from $T_1$.
2) Remove all leaves from $T_2$ that are also in $s$.
3) The offspring subtree $T'_2$ is obtained by regrafting $s$ to an edge randomly chosen from $T_2$.

The second offspring, denoted as $T'_2$, is created in a similar way: prune a subtree from $T_2$ and regraft it in $T_1$. Fig. 3 illustrates this operator.

![Fig. 3. Example of the crossover operator.](image)

E. Mutation Operator

There are three well-known topological modifications used in phylogenetic inference [11]: Nearest Neighbor Interchange (NNI), Sub-tree Pruning and Regrafting (SPR) and Tree Bisection and Reconnection (TBR). NNI was employed in PhyloMOEA, since it performs minimal tree modifications. This operator carries out the following steps:

1) Choose an interior branch whose connected nodes $i$, $j$ define two pairs of neighbors: $w$, $x$ adjacent to $i$ ($w, x \neq j$) and $y$, $z$ adjacent to $j$ ($y, z \neq i$).
2) Execute a swap of two nodes taken from each pair of neighbors.

Fig. 4 illustrates the NNI operator. The mutation operator also modifies branch lengths in order to improve the tree likelihood value. A branch length is multiplied by a factor obtained from a gamma distribution [18]. In each mutation, some branch lengths, chosen at random, are modified.

Branch lengths from trees in the final population are optimized using a non-decreasing Newton-Raphson method described by Yang [31]. Due to this optimization being very time-consuming, it is applied only after a PhyloMOEA execution. Fig. 5 shows the PhyloMOEA algorithm.
Algorithm: PhyloMOEA

begin
1. Create an initial population $P_1$ containing $N$ solutions
2. Perform non-dominated Sorting in $P_1$
3. Calculate crowding distance values of $P_1$
4. Apply selection, crossover and mutation operators in $P_1$ and generate a new population $Q_1$

foreach generation $t = 2, \ldots, n$ do
5. Perform non-dominated sorting in $R = P_t \cup Q_t$
6. Calculate crowding distance values of $R$
7. Calculate Pareto frontiers $F_1, F_2, \ldots, F_j$ from $R$
8. Store the $N$ best solutions from $F_k$ in $P_{t+1}$, $F_k \leq N$, $k = 1 \ldots j$
9. Create a new population $Q_{t+1}$ by applying selection, crossover and mutation operators in $P_{t+1}$
10. Perform branch length optimization of solutions in $P_n$.
end

VI. Results and Discussion

PhyloMOEA was tested using four nucleotide data sets. The rbcL,55 dataset comprises 55 sequences (1314 sites) of the rbcL chloroplast gene from green plants [18]. The mtDNA,186 dataset contains 186 human mitochondrial DNA sequences (16608 sites) taken from The Human Mitochondrial Genome Database (mtDB) [32]. The RDP II,218 dataset comprises 218 prokaryotic RNA sequences (4182 sites) taken from the Ribosomal Database Project II [33]. Finally, the ZILLA,500 dataset includes 500 rbcL sequences (1428 sites) from plant plastids [16].

Maximum parsimony and likelihood analyzes were performed in the four datasets using programs NONA [34] and RAXML-V [17], respectively. These programs include sophisticated heuristics that produce satisfactory results quickly. Table I shows the parsimony and likelihood scores of the results obtained from these programs.

Trees in the initial population were generated from bootstrap data applied to each dataset. These data was obtained using the program PHYML [16], which performs bootstrap analysis using the BIONJ algorithm [35]. The parsimony and likelihood scores of these trees are close to the scores shown in Table I. However, for RDP II,218 and ZILLA,500 datasets, bootstrap tree scores are not close enough to the scores obtained by NONA and RAXML-V. Consequently, it slows down the PhyloMOEA’s convergence. In order to overcome this drawback, solutions from Table I are included in the initial population.

Table II shows the parameters of PhyloMOEA used for the experiments. It can be noted that ZILLA,500 dataset requires the largest number of generations and population size. Furthermore, the HKY85 [36] nucleotide model was used in likelihood calculations since it is often used in the literature [16]–[19].

VI. Results and Discussion

PhyloMOEA was tested using four nucleotide data sets. The rbcL,55 dataset comprises 55 sequences (1314 sites) of the rbcL chloroplast gene from green plants [18]. The mtDNA,186 dataset contains 186 human mitochondrial DNA sequences (16608 sites) taken from The Human Mitochondrial Genome Database (mtDB) [32]. The RDP II,218 dataset comprises 218 prokaryotic RNA sequences (4182 sites) taken from the Ribosomal Database Project II [33]. Finally, the ZILLA,500 dataset includes 500 rbcL sequences (1428 sites) from plant plastids [16].

Maximum parsimony and likelihood analyzes were performed in the four datasets using programs NONA [34] and RAXML-V [17], respectively. These programs include sophisticated heuristics that produce satisfactory results quickly. Table I shows the parsimony and likelihood scores of the results obtained from these programs.

Trees in the initial population were generated from bootstrap data applied to each dataset. These data was obtained using the program PHYML [16], which performs bootstrap analysis using the BIONJ algorithm [35]. The parsimony and likelihood scores of these trees are close to the scores shown in

VI. Results and Discussion

PhyloMOEA was tested using four nucleotide data sets. The rbcL,55 dataset comprises 55 sequences (1314 sites) of the rbcL chloroplast gene from green plants [18]. The mtDNA,186 dataset contains 186 human mitochondrial DNA sequences (16608 sites) taken from The Human Mitochondrial Genome Database (mtDB) [32]. The RDP II,218 dataset comprises 218 prokaryotic RNA sequences (4182 sites) taken from the Ribosomal Database Project II [33]. Finally, the ZILLA,500 dataset includes 500 rbcL sequences (1428 sites) from plant plastids [16].

Maximum parsimony and likelihood analyzes were performed in the four datasets using programs NONA [34] and RAXML-V [17], respectively. These programs include sophisticated heuristics that produce satisfactory results quickly. Table I shows the parsimony and likelihood scores of the results obtained from these programs.

Trees in the initial population were generated from bootstrap data applied to each dataset. These data was obtained using the program PHYML [16], which performs bootstrap analysis using the BIONJ algorithm [35]. The parsimony and likelihood scores of these trees are close to the scores shown in Table I. However, for RDP II,218 and ZILLA,500 datasets, bootstrap tree scores are not close enough to the scores obtained by NONA and RAXML-V. Consequently, it slows down the PhyloMOEA’s convergence. In order to overcome this drawback, solutions from Table I are included in the initial population.

Table II shows the parameters of PhyloMOEA used for the experiments. It can be noted that ZILLA,500 dataset requires the largest number of generations and population size. Furthermore, the HKY85 [36] nucleotide model was used in likelihood calculations since it is often used in the literature [16]–[19].

### Table I

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NONA</th>
<th>RAXML-V</th>
</tr>
</thead>
<tbody>
<tr>
<td>rbcL,55</td>
<td>4874</td>
<td>-24627.848</td>
</tr>
<tr>
<td>mtDNA,186</td>
<td>2438</td>
<td>-41049.768</td>
</tr>
<tr>
<td>RDP II,218</td>
<td>41534</td>
<td>-170831.121</td>
</tr>
<tr>
<td>ZILLA,500</td>
<td>16219</td>
<td>-87361.484</td>
</tr>
</tbody>
</table>

Table I. However, for RDP II,218 and ZILLA,500 datasets, bootstrap tree scores are not close enough to the scores obtained by NONA and RAXML-V. Consequently, it slows down the PhyloMOEA’s convergence. In order to overcome this drawback, solutions from Table I are included in the initial population.

Table II shows the parameters of PhyloMOEA used for the experiments. It can be noted that ZILLA,500 dataset requires the largest number of generations and population size. Furthermore, the HKY85 [36] nucleotide model was used in likelihood calculations since it is often used in the literature [16]–[19].

### Table II

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generations</td>
<td>500 (rbcL,55, mtDNA,186, and RDP II,218)</td>
</tr>
<tr>
<td>Population size</td>
<td>2000 (ZILLA,500)</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.8</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.05</td>
</tr>
<tr>
<td>Mutation operator</td>
<td>NNI</td>
</tr>
<tr>
<td>Evolution model</td>
<td>HKY85</td>
</tr>
</tbody>
</table>

Due to the stochastic nature of GAs, PhyloMOEA was executed 20 times for each dataset. At the end of a PhyloMOEA execution, duplicate tree topologies are removed from the final population. Finally, the Pareto optimal solutions are calculated, although this may eliminate promising topologies from the perspective of parsimony criterion. If two solutions have an equal parsimony score, only the solution with the best likelihood remains in the Pareto-set. Thus, all non-duplicated topologies, entitled Final Solutions, are maintained.

Table III presents a summary of the experiment results. It shows the maximum, average and standard deviation of the number of Pareto and Final solutions for all executions. Moreover, this table shows the best score, average score and standard deviation for the maximum parsimony and maximum likelihood criteria for all runs.

The values in bold in Table III shows the parsimony and likelihood scores improved by PhyloMOEA when compared with scores from Table I. In the case of the parsimony criterion, only the mtDNA,186 score was improved. On the
other hand, all likelihood scores were slightly increased, except for the 500_ZILLA dataset, where a better improvement was reached.

Figs. 6, 7, 8 and 9 show the Pareto fronts obtained in one PhyloMOEA execution for \( rbcL \), \( mtDNA \), \( RDP II \) and ZILLA datasets, respectively. These figures also show Final Solutions near the Pareto front. Due to the parsimony score being an integer value, the resulting Pareto front is a discontinuous set of points connected by lines. The two extreme points from the frontier represent the maximum parsimony and maximum likelihood trees found by PhyloMOEA. If both points are close to one another, a reduced number of intermediate solutions is expected. This is the case of \( rbcL \) and \( mtDNA \) datasets, as illustrated in Figs. 6 and 7. Moreover, Table III shows the small size of the Pareto front for both datasets. On the other hand, extreme points in \( RDP II \) and ZILLA datasets are distant one from another. Thus, there are a greater number of intermediate solutions, as shown in Figs. 8 and 9 and in Table III. Nevertheless, for all datasets, PhyloMOEA was able to find a relatively large number of Final Solutions.

When a set of trees is obtained from a phylogenetic analysis, it is useful to compare these solutions using a statistical test. The Shimodaira-Hasegawa test (SH test) [37] was used in order to evaluate the Pareto optimal and Final Solutions found by PhyloMOEA. The SH-test calculates a \( P \)-value for each solution, which indicates if a tree is significantly worse than the best scored one according to a criterion. If a tree has a \( P \)-value lower than a given bound (usually 0.05), it can be rejected. The SH-test was performed for parsimony and likelihood criteria using programs PHYLIP [26] and PAML [38], respectively.

Table IV shows summary results from the SH-test applied to the best PhyloMOEA's execution in each dataset. The number of non-rejected (\( P \geq 0.05 \)) and rejected (\( P < 0.05 \)) trees according to parsimony and likelihood criteria for Pareto and Final Solutions are shown. For the \( rbcL \) dataset, it can be seen that none of the Pareto solutions were rejected for the SH-test applied to both criteria. This is due to extreme solutions in Pareto front having their parsimony and likelihood scores close, and therefore intermediate solutions cannot be rejected. In the case of the \( mtDNA \) dataset, only parsimony scores of extreme points are close while likelihood scores are distant. Consequently, the SH-test applied to the parsimony criterion does not reject any solution. However, four solutions are rejected for likelihood criterion. On the other hand, extreme solutions scores for \( RDP II \) and ZILLA datasets are distant. Thus, SH-test rejects a number of Pareto solutions for parsimony and likelihood criteria.

In the case of the Final Solutions, the SH-test applied to parsimony and likelihood criteria rejects approximately two thirds of the solutions for \( rbcL \), \( RDP II \) and ZILLA datasets. The only exception is the SH-test using the likelihood criterion for the \( RDP II \) dataset where most solutions are rejected. On the other hand, the SH-test for parsimony criteria does not reject any of the Final solutions from the \( mtDNA \) dataset. It reveals that parsimony scores for Final Solutions are close to the best parsimony score. The likelihood scores of Final Solutions from the \( mtDNA \) dataset are also close to the maximum likelihood score, but in this case the proportion of rejected solutions are greater.

We pointed out that the SH-test was designed to be applied for one criterion, i.e. this is not a multi-criteria test. However, the SH-test shows that some of the Pareto optimal solutions are not significantly worse than the best trees resulting from a separate analysis. Thus, PhyloMOEA was able to find intermediate solutions that are consistent with the best solutions obtained from the parsimony and likelihood criteria.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a MOEA to solve the phylogenetic inference problem using parsimony and likelihood criteria. This approach was motivated by the literature in area [1]–[3], [28], which points out that various phylogenetic inference methods leads to inconsistent solutions. This fact was verified for all datasets analyzed in the experiments where parsimony and likelihood criteria leads to different trees. The proposed algorithm, called PhyloMOEA, was able to find a set of trees that represents a trade-off between these criteria. A SH-test applied to Pareto and Final Solutions found by PhyloMOEA indicates that some alternative solutions are consistent with criteria used.

Despite the relevant results found by PhyloMOEA, there are aspects that should be addressed in order to improve the algorithm and corresponding results:

- PhyloMOEA requires several hours to find acceptable Pareto-solutions if initial trees are poorly estimated. This problem can be improved using advanced genetic operators that take into account local search strategies [16], [17]. PhyloMOEA's performance is also decreased by the likelihood calculation, which is computationally intensive. There are some techniques that address this problem [17], [39], [40];
- PhyloMOEA does not optimize parameters of the evolution model employed in the likelihood calculation. It uses approximated values for these parameters. These values can be optimized when the algorithm is running [18];
- PhyloMOEA uses a simple cost matrix to calculate the parsimony score. There is some parsimony criteria that uses more complex cost matrix and its use may improve results [11];
- The likelihood calculation performed by PhyloMOEA does not consider the rate heterogeneity among sites. In real datasets, sites frequently evolve at different rates. When rate heterogeneity is employed, the accuracy of the likelihood analysis is often improved [41].
- This research has not investigated metrics for convergence and diversity of the obtained Pareto front. Measurements for convergence are difficult to obtain since the Pareto front is unknown in this case. However, various diversity metrics found in the literature [21] can be employed.
TABLE III
Summary of experiments' results.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Pareto Trees</th>
<th>Number of Final Trees</th>
<th>Parsimony Tree Scores</th>
<th>Likelihood Tree Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max.</td>
<td>Average ±σ</td>
<td>Best</td>
<td>Average ±σ</td>
</tr>
<tr>
<td><em>rbcL</em></td>
<td>55</td>
<td>7.05 ±1.39</td>
<td>4874</td>
<td>4874.00 ±0.00</td>
</tr>
<tr>
<td><em>mtDNA</em></td>
<td>186</td>
<td>9.05 ±1.23</td>
<td>54</td>
<td>4895 ±2.61</td>
</tr>
<tr>
<td>218<em>RDP II</em></td>
<td>35</td>
<td>28.75 ±2.97</td>
<td>85</td>
<td>77.40 ±4.15</td>
</tr>
<tr>
<td>500<em>ZILLA</em></td>
<td>24</td>
<td>18.50 ±2.52</td>
<td>121</td>
<td>102.40 ±7.99</td>
</tr>
</tbody>
</table>

TABLE IV
Summary of SH-test results.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><em>rbcL</em></td>
<td>10</td>
<td>0</td>
<td>16</td>
<td>37</td>
<td>17</td>
<td>36</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td><em>mtDNA</em></td>
<td>8</td>
<td>0</td>
<td>4</td>
<td>37</td>
<td>8</td>
<td>22</td>
<td>23</td>
<td>4</td>
</tr>
<tr>
<td>218<em>RDP II</em></td>
<td>10</td>
<td>25</td>
<td>6</td>
<td>29</td>
<td>21</td>
<td>57</td>
<td>11</td>
<td>67</td>
</tr>
<tr>
<td>500<em>ZILLA</em></td>
<td>12</td>
<td>9</td>
<td>14</td>
<td>7</td>
<td>27</td>
<td>79</td>
<td>29</td>
<td>77</td>
</tr>
</tbody>
</table>

Since PhyloMOEA produces a set of trees, it is possible to calculate branch support values and consensus trees in each dataset. These calculations can be used to compare PhyloMOEA’s results with other methods, that produce solution sets, such as a bootstrap analysis [42] or Bayesian inference [43].

To sum up, preliminary results have shown that PhyloMOEA can make relevant contributions to phylogenetic inference. Moreover, there are several aspects that can be investigated to improve the current approach.
REFERENCES