Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2007)

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.

The authors thank the State of Sao Paulo Research Foundation (FAPESP) for the financial support provided (Grant Nº 01/13846-0).

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

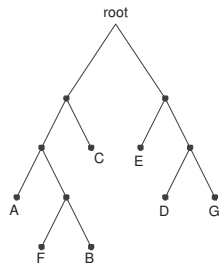

Fig. 1. A rooted tree.

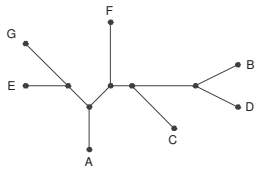

Fig. 2. An unrooted tree.

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 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:

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

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

$$ps_j(T) = \sum_{(v,u)\in E(T)} C(v_j, u_j), \tag{2}$$

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 is calculated from the product of partial likelihoods from all sites:

$$L = \prod_{i=1}^{N} L_i, \tag{3}$$

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:

$$\ln L = \sum_{i=1}^{n} L_i \tag{4}$$

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 of 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 be better than the average fitness of the previous population. This process is repeated until a stop criterion has been reached. Thus, the fitness landscape in combination with the selection leads GAs towards an optimal or a near-optimal answer. 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 $\mathbf{x}$ dominates a solution $\mathbf{y}$ if $\mathbf{x}$ is not worse than $\mathbf{y}$ 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 of 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 $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_j$. The first frontier ($\mathcal{F}_1$) is formed by non-dominated solutions from R. Solutions in $\mathcal{F}_1$ are removed from R and the remaining solutions are employed to calculate the next set of non-dominated solutions, denoted by $\mathcal{F}_2$. This process is repeated in order to find $\mathcal{F}_3$, 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_1' 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.

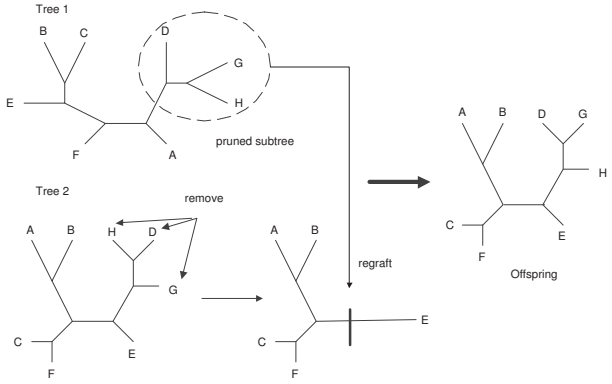

Fig. 3. Example of the crossover operator.

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.

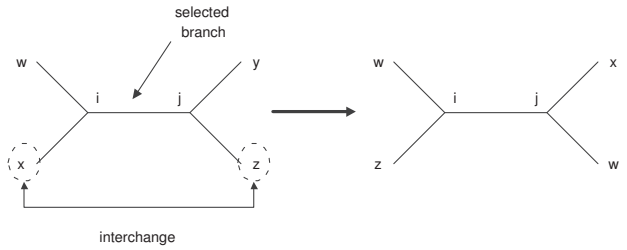

Fig. 4. Example of NNI mutation operator.

Algorithm: PhyloMOEA

begin

Create an initial population P_1 containing N solutions
Perform non-dominated Sorting in P_1
Calculate crowding distance values of P_1
Apply selection, crossover and mutation operators in P_1 and generate a new population Q_1
foreach *generation* $t = 2, \ldots, n$ **do**
    Perform non-dominated sorting in $R = P_t \cup Q_t$
    Calculate crowding distance values of R
    Calculate Pareto frontiers $\mathcal{F}_1, \mathcal{F}_2, \ldots \mathcal{F}_j$ from R
    Store the N best solutions from $\mathcal{F}_k$ in P_{t+1},$|\mathcal{F}_k| \leq N, \quad k = 1 \ldots l$
    Create a new population Q_{t+1} by applying selection, crossover and mutation operators in P_{t+1}
Perform branch length optimization of solutions in P_n.

end

Fig. 5. PhyloMOEA algorithm.

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 $RDPII\_218$ 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 $RDPII\_218$ 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 I
Parsimony and Likelihood results found by NONA and RAxML-V.

Dataset	NONA		RAxML-V	
	Pars.	Likelihood	Pars.	Likelihood
$rbcL\_55$	4874	-24627.848	4894	-24583.331
$mtDNA\_186$	2438	-41049.768	2450	-40894.550
$RDPII\_218$	41534	-170831.121	42631	-156595.873
$ZILLA\_500$	16219	-87361.484	16276	-86993.826

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
Parameters used by PhyloMOEA in the experiments.

Parameter	Value
Generations	500 ($rbcL\_55, mtDNA\_186$, and $RDPII\_218$)
	2000 ($ZILLA\_500$)
Population size	50 ($rbcL\_55, mtDNA\_186$, and $RDPII\_218$)
	100 ($ZILLA\_500$)
Crossover rate	0.8
Mutation rate	0.05
Mutation operator	NNI
Evolution model	HKY85

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 $rbcL\_55$, $mtDNA\_186$, $RDPII\_218$ and $ZILLA\_500$ 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 $rbcL\_55$ and $mtDNA\_186$ 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 $RDPII\_218$ and $ZILLA\_500$ 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 $rbcL\_55$ 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 $mtDNA\_186$ 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 $RDPII\_218$ and $ZILLA\_500$ 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 $rbcL\_55$, $RDPII\_218$ and $ZILLA\_500$ datasets. The only exception is the SH-test using the likelihood criterion for the $RDPII\_218$ dataset where most solutions are rejected. On the other hand, the SH-test for parsimony criteria does not reject most of the Final solutions from the $mtDNA\_186$ dataset. It reveals that parsimony scores for Final Solutions are close to the best parsimony score. The likelihood scores of Final Solutions from the $mtDNA\_186$ 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.

Dataset	Number of Pareto Trees		Number of Final Trees		Parsimony Tree Scores		Likelihood Tree Scores	
	Max.	Average $\pm\sigma$	Max.	Average $\pm\sigma$	Best	Average $\pm\sigma$	Best	Average $\pm\sigma$
$rbcL\_55$	10	7.05 $\pm$1.39	54	48.20 $\pm$3.00	4874	4874.00 $\pm$0.00	**-24583.330**	**-24583.330** $\pm$0.00
$mtDNA\_186$	12	9.05 $\pm$1.23	55	48.95 $\pm$2.61	**2436**	**2437.10** $\pm$0.64	**-40894.343**	**-40894.528** $\pm$0.06
218_RDPII	35	28.75 $\pm$2.97	85	77.40 $\pm$4.15	41534	41534.00 $\pm$0.00	**-156595.850**	**-156595.850** $\pm$0.00
500_ZILLA	24	18.50 $\pm$2.52	121	102.40 $\pm$7.99	16219	16219.00 $\pm$0.00	**-86991.649**	**-86993.561** $\pm$0.66

TABLE IV

SUMMARY OF SH-TEST RESULTS.

Dataset	Pareto Trees				Final Trees			
	SH-test Parsimony		SH-test Likelihood		SH-test Parsimony		SH-test Likelihood	
	Non-Rej.	Rej.	Non-Rej.	Rej.	Non-Rej.	Rej.	Non-Rej.	Rej
$rbcL\_55$	10	0	10	0	16	37	17	36
$mtDNA\_186$	8	0	4	4	37	8	22	23
$RDPII\_218$	10	25	6	29	21	57	11	67
$ZILLA\_500$	12	9	14	7	27	79	29	77

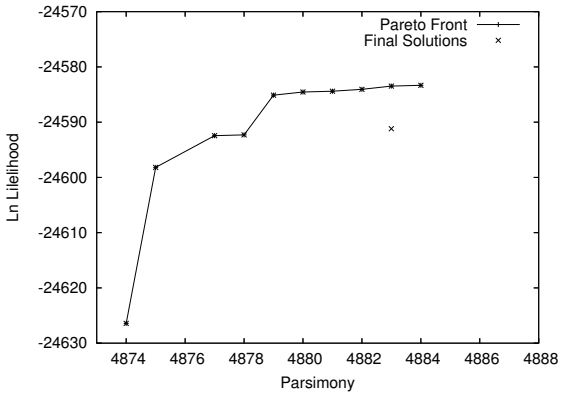

Fig. 6. Final Solutions and Pareto front for $rbcL\_55$ dataset.

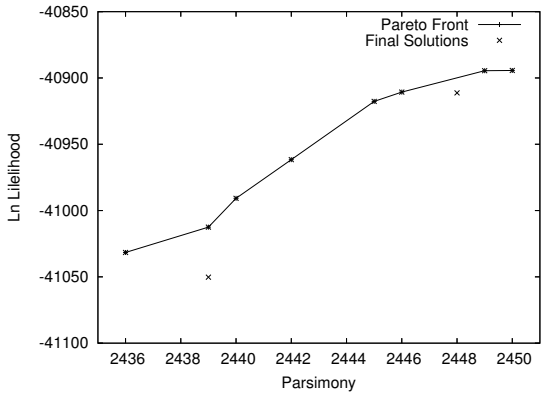

Fig. 7. Final Solutions and Pareto front for $mtDNA\_186$ dataset.

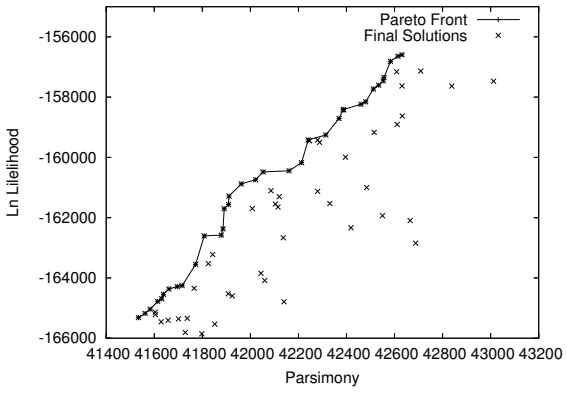

Fig. 8. Final Solutions and Pareto front for $RDPII\_218$ dataset.

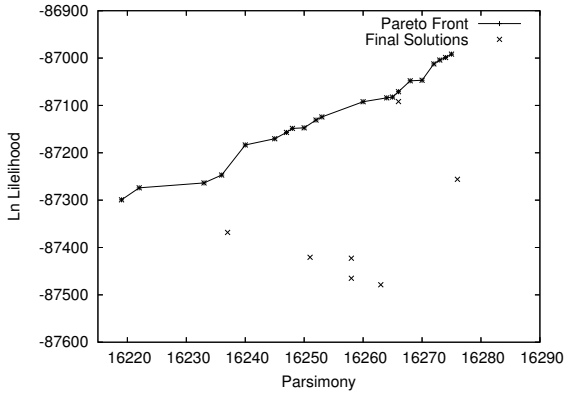

Fig. 9. Final Solutions and Pareto front for $ZILLA\_500$ dataset.

- 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

[1] J. Huelsenbeck, "Performance of Phylogenetic Methods in Simulation," *Systematic Biology*, vol. 44, p. 17–48, 1995.

[2] M. Kuhner and J. Felsenstein, "A Simulation Comparison of Phylogeny Algorithms under Equal and Unequal Evolutionary Rate," *Molecular Biology and Evolution*, vol. 11, pp. 459–468, 1994.

[3] Y. Tateno, N. Takezaki, and M. Nei, "Relative Efficiences of the Maximum-Likelihood, Neighbor-Joining, and Maximum Parsimony Methods when Substitution Rate Varies with Site," *Molecular Biology and Evolution*, vol. 11, pp. 261–267, 1994.

[4] A. Rokas, B. Wiliams, N. King, and S. Carroll, "Genome-Scale Approaches to Resolving Incongrounce in Molecular Phylogenies," *Nature*, vol. 425, no. 23, p. 798–804, 2003.

[5] J. Handl, D. Kell, and J. Knowles, "Multiobjective Optimization in Computational Biology and Bioinformatics," *IEEE Transactions on Computational Biology and Bioinformatics*, 2006, to appear.

[6] L. Poladian and L. Jermiin, "Multi-Objective Evolutionary Algorithms and Phylogenetic Inference with Multiple Data Sets," *Soft Computing*, vol. 10, no. 4, p. 359–368, 2006.

[7] W. Fitch, "Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology," *Systematic Zoology*, vol. 20, no. 4, p. 406–416, 1972.

[8] J. Felsenstein, "Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach," *Journal of Molecular Evolution*, vol. 17, p. 368–376, 1981.

[9] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," Indian Institute of Technology, Kanpur, India, KanGAL report 200001, 2000.

[10] J. Felsenstein, *Inferring Phylogenies*. Sunderland, Massachusetts: Sinauer, 2004.

[11] D. Swofford, G. Olsen, P. Waddell, and D. Hillis, "Phylogeny Reconstruction," in *Molecular Systematics*, 3rd ed. Sinauer, 1996, ch. 11, p. 407–514.

[12] N. Saitou and M. Nei, "The Neighbor-Joining Method: A New Method for Reconstructing Phylogenetic Trees," *Molecular Biology and Evolution*, vol. 4, no. 4, p. 406–425, 1987.

[13] L. Cavalli-Sforza and A. Edwards, "Phylogenetic Analysis: Models and Estimation Procedures," *Evolution*, vol. 21, no. 3, p. 550–570, 1967.

[14] D. Sankoff, "Simultaneous Solution of the RNA Folding, Alignment and Proto-Sequence Problems," *SIAM Journal on Applied Mathematics*, vol. 45, no. 5, pp. 810–825, 1985.

[15] P. Goloboff, "Methods for faster parsimony analysis," *Cladistics*, vol. 12, no. 3, p. 199–220, 1996.

[16] S. Guindon and O. Gascuel, "A Simple, Fast, and Accurate Algorithm to Estimate Large Phylogenies by Maximum Likelihood," *Systematic Biology*, vol. 5, no. 52, p. 696–704, 2003.

[17] A. Stamatakis and H. Meier, "New Fast and Accurate Heuristics for Inference of Large Phylogenetic Trees," in *18th IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS2004)*, 2004.

[18] P. O. Lewis, "A Genetic Algorithm for Maximum-Likelihood Phylogeny Inference Using Nucleotide Sequence Data," *Molecular Biology and Evolution*, vol. 15, no. 3, p. 277–283, 1998.

[19] A. R. Lemmon and M. C. Milinkovitch, "The Metapopulation Genetic Algorithm: An Efficient Solution for the Problem of Large Phylogeny Estimation," in *Proceedings of the National Academy of Sciences*, vol. 99, no. 16, 2002, p. 10516–10521.

[20] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1989.

[21] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. New York: John Wiley & Sons, 2001.

[22] K. Katoh, K. Kuma, and T. Miyata, "Genetic Algorithm-Based Maximum-Likelihood Analysis for Molecular Phylogeny," *Journal of Molecular Evolution*, vol. 53, p. 477–484, 2001.

[23] C. B. Congdon and K. J. Septor, "Phylogenetic trees using evolutionary search: Initial progress in extending GAPHYL to work with genetic data," in *Congress on Evolutionary Computation (CEC-2003)*, R. Sarker, R. Reynolds, H. Abbass, K. C. Tan, B. McKay, and T. Gedeon, Eds., vol. 1. IEEE Press, 2003, p. 320–326.

[24] A. Moilanen, "Searching for Most Parsimonious Trees with Simulated Evolutionary Optimization," *Cladistics*, vol. 15, p. 39–50, 1999.

[25] C. Cotta and P. Moscato, "Inferring Phylogenetic Trees Using Evolutionary Algorithms," in *Parallel Problem Solving From Nature VII*, J. Merelo, Ed. Springer-Verlag, 2002, p. 720–729.

[26] J. Felsenstein, "PHYLIP (Phylogeny Inference Package)," 2000.

[27] D. Swofford, "PAUP* Phylogenetic Analisys Using Parsimony," 2000, cSIT Florida State University.

[28] L. Jin and M. Nei, "Limitations of the Evolutionary Parsimony Method of Phylogenetic Analysis," *Molecular Biology and Evolution*, vol. 7, pp. 82–102, 1990.

[29] J. Adachi and M. Hasegawa, "MOLPHY version 2.3. Programs for Molecular Phylogenetics Based on Maximum Likelihood. in ishiguro," in *Computer Science Monographs*, M. Ishiguro, G. Kitagawa, Y. Ogata, H. Takagi, Y. Tamura, and T. Tsuchiya, Eds., The Institute of Statistical Mathematics, Tokyo, 1996, no. 28.

[30] M. Forster, A. Pick, M. Raitner, and C. Bachmaier, *GTL - Graph Template Library Documentation*, University of Pasdau, 2004. [Online]. Available: http://infosun.fmi.uni-passau.de/GTL/

[31] Z. Yang, "Maximum-Likelihood Estimation of Phylogeny from DNA Sequences when Substitution Rates Differ over Sites," *Molecular Biology and Evolution*, vol. 10, no. 6, p. 1396–1401, 1993.

[32] M. Ingman and U. Gyllensten, "mtDB: Human Mitochondrial Genome Database, a Resource for Population Genetics and Medical Sciences," *Nucleic Acids Research*, vol. 34, pp. D749–D751, 2006.

[33] J. Cole, B. Chai, R. Farris, Wang, S. Kulam, D. McGarrell, G. Garrity, and J. Tiedje, "The Ribosomal Database Project (RDP-II): Sequences and Tools for High-throughput rRNA Analysis," *Nucleic Acids Research*, vol. 33, pp. D294–D296, 2005.

[34] P. Goloboff, "NONA (no name) ver. 2," Tucuman, Argentina, 1999, published by the author. [Online]. Available: http://www.cladistics.com/aboutNona.htm

[35] O. Gascuel, "BIONJ: An Improved Version of the NJ Algorithm Based on a Sample Model of Sequence Data," *Molecular Biology and Evolution*, vol. 14, no. 7, p. 685–695, 1997.

[36] M. Hasegawa, H. Kishino, and T. Yano, "Dating of the Human–Ape Splitting by a Molecular Clock of Mitochondrial DNA," *Journal of Molecular Evolution*, vol. 22, p. 160–174, 1985.

[37] H. Shimodaira and M. Hasegawa, "Likelihood-Based Tests of Topologies in Phylogenetics," *Molecular Biology and Evolution*, vol. 16, no. 8, p. 1114–1116, 1999.

[38] Z. Yang, "PAML: A Program Package for Phylogenetic Analysis by Maximum Likelihood." *Computer Applications in Biosciences*, vol. 13, no. 5, pp. 555–6, 1997.

[39] S. K. Pond and S. Muse, "Column Sorting: Rapid Calculation of the Phylognetic Likelihood Function," *Systematic Biology*, vol. 53, no. 5, pp. 685–592, 2004.

[40] B. Larget and D. Simon, "Faster likelihood calculations on trees," Department of Mathematics and Computer Science. Duquesne University, Tech. Rep., 1998.

[41] Z. Yang, "Maximum Likelihood Estimation on Large Phylogenies and Analysis of Adaptative Evolution in Human Influenza Virus A," *Journal of Molecular Evolution*, vol. 51, no. 5, p. 423–432, 2000.

[42] J. Felsenstein, "Confidence Limits on Phylogenies: An Approach Using the Bootstrap," *Evolution*, vol. 39, no. 4, pp. 783–791, 1985.

[43] Z. Yang and B. Rannala, "Bayesian Phylogenetic Inference Using DNA Sequences: A Markov Chain Monte Carlo Method," *Molecular Biology and Evolution*, vol. 14, no. 7, pp. 717–724, 1997.