IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 27, 2019, accepted December 10, 2019, date of publication December 16, 2019,
date of current version January 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2959783

Parallel MOEA Based on Consensus and Membrane Structure for Inferring Phylogenetic Reconstruction

QIANQIAN ZHANG[1], JUN ZHANG[2], YUE ZHONG[1], CONGMING YE[1], AND XIAOPING MIN[1]

[1]Department of Computer Science, Xiamen University, Xiamen 361005, China
[2]Rehabilitation Department, Heilongjiang Province Land Reclamation Headquarters General Hospital, Harbin 150088, China

Corresponding author: Xiaoping Min (mxp@xmu.edu.cn)

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61772441, 61872309, 61922020, 61425002, 61673328, 61872007); the national key R&D program of China (2017YFE0130600); Project of marine economic innovation and development in Xiamen (No. 16PFW034SF02); Natural Science Foundation of Fujian Province (No. 2017J01099); Fundamental Research Funds for the Central Universities (63192616).

ABSTRACT In recent years, inferring phylogenies has attracted lots of attention in both academic community and various application fields. Phylogenetic inference usually consists of a couple of evolutionary relationships, which can be represented as a phylogenetic tree. The phylogenetic reconstruction problem can be defined as an optimization problem, targeting at finding the most eligible tree among all possible topologies according to a selected criterion. Since the combinatorial number of possible topologies exceeds tolerance, various heuristic and metaheuristic methods have been proposed to find approximate solutions according to the selected criterion. However, different criterions are based on different principle and conflict with each other basically. In this line, scholars has proposed multi-objective evolutionary algorithm (MOEA) based on diverse criteria. Nevertheless, MOEA has suffered unbearable time consumption due to its inherent drawbacks of computational complexity and convergence. By studying the independence between the subpopulations in each time-consuming step of MOEA, the steps without global information can be designed to be executed in parallel, which can fundamentally address computational problems. Effective parallel algorithms designed with the characteristics of modern multicore clusters can solve such problems. In this sense, we propose a parallelized multi-objective evolutionary algorithm (MOEA-MC) by deploying on Spark, which added consensus into evolutionary algorithm to improve the quality of convergence and used membrane structure to keep equal solutions under different weights. In order to assess the performance achieved by the proposal, we have performed comparison among different methods on three real-world datasets separately. The results have certified that the solutions derived from MOEA-MC are superior to traditional methods in all studied datasets. And parallelized MOEA-MC can get dominant position and optimal Pareto-frontier simultaneously within minimal runtime.

INDEX TERMS Consensus, multi-objective evolutionary algorithm, membrane structure, phylogenetic reconstruction, parallel algorithm.

I. INTRODUCTION

Biological research has gradually attracted the attention of scholars with the explosive growth of the amount of genomic data published in the past few decades. In particular, phylogenetic reconstruction is one of the main research areas of bioinformatics. Phylogenetic inference consists of a series of evolutionary relationships, which usually be represented as a phylogenetic tree. Phylogenetic reconstruction can be used to describe the evolutionary relationships between molecules, which can promote the research of biomedical, genetic prediction, and economical crop. For example, Zhang [1] constructed Arabidopsis and rice AT-hook proteins into phylogenetic trees which found that AT-hook genes can be divided into five subfamilies with similar structures and characteristics. The publication shows the evolutionary

The associate editor coordinating the review of this manuscript and approving it for publication was Quan Zou.

relationships among different organisms which can helps predict the function of rice genes. In addition, the next-generation sequencing revolution has brought unprecedented growth in phylogenetic analysis data sets. And phylogenetic reconstruction devoted to reconstruct a biological phylogenetic tree that explaining the evolutionary relationship among a given biological sequence file.

The various bioinformatics issues involve complex optimizations, and biologists are committed to finding accurate explanations based on biological principles. This issue motivated the development of effective algorithm design to address current requirements. In this sense, using bio-inspired meta-heuristics to overcome computational challenges [2] have become an increasingly popular method. Phylogenetic reconstruction based on evolutionary and bio-inspired algorithms can be categorized as an optimization problem that finds the best topology among all possible trees based on the selected objective function or criteria. Huelsenbeck [3] explained that the trees which reconstruct according to different criteria may conflict with others, even if they owned the same input. Rokas *et al.* [4] also pointed out that the selection of criteria has a great influence on the final results. Accordingly, Handl *et al.* [5] proposed and recommended the application of multi-objective optimization. As [6] mentioned, multi-objective optimization has follow ascendants when compared with the single-objective method : 1) minimize the local minimum and the probability of stagnation in the gradient-free region; 2) reduce the noise impact of the data; 3) introduce multiple sources that conflict with each other which can meet multiple standards concurrently. Therefore, transform phylogeny inference into multi-objective optimization problem (MOP) [7], [8] has taken the mainstream stage. The development of MOP will bring dawn to biologists. Tree generated by MOP are not only supported by different biological principles, but also have high-quality topologies from the perspective of each objective function. The complexity of evolutionary inference has been increased with the new perspective, which has inspired researchers to conduct original research based on heuristic algorithms [9]. Since the emergence of the multi-objective evolutionary algorithms(MOEA), problems involving complex and diverse optimization have transform into finding accurate solutions according several biological principles [10].

According to different selection mechanisms, MOEA can be divided into the following categories: aggregation functions; population-based approaches; Pareto-based approaches. Most of the current considerations are based on Pareto, and the process of a multi-objective evolutionary algorithm based on Pareto is as follows: First, generate an initial population P, and then selected an evolutionary algorithm (such as a genetic algorithm) to perform evolutionary operations (such as crossover, mutation, and selection) on P to obtain a new evolutionary group R. Then construct the non-dominated set NDSet of $P \cup R$. If the current non-dominated set NDSet is greater or less than the preset size of the non-dominated set N, it is necessary to adjust. On the other hand, the NDSet also need meet the distribution requirement. If met the termination condition, it ends, otherwise copied the individuals in the NDSet to P and the next round of evolution is continued. Pareto-based approaches are relies too much on the selection of shared parameters and generate greater selection pressure, which leads to immature convergence. In addition, each iterations needs to calculate the fitness values of all individuals in the current population, thereby increasing the execution time of algorithm implementation. The two key issues in the implementation of MOEA based on Pareto are: 1) How to make the population search towards the Pareto frontier as soon as possible, that is, the convergence of the population. 2) How to obtain a non-inferior solution with uniform distribution on the Pareto frontier, that is, the diversity of the population. Such as NSGA-II [11] proposed a fast non-dominated sorting, which uses the crowded distance to measure the distribution of solutions and operate selection, but it is complicated to calculate crowded distance. In addition, the computational complexity of NSGA-II is too high in high-dimensional multi-objective problem. MOEA/D [12] converts a multi-objective optimization problem into multiple scalar quantum problems, and each sub-problem consists of a uniformly distributed weight vector. Once a new solution is generated, the solution near the sub-problem is replaced based on the aggregate function. However, evenly distributed weight vector on the unit hyperplane is unable to guarantee uniform distribution of the final solution. These inherent natures have caused the following defects. First, the difficulty of solving the objective function greatly extended the execution time. Second, the convergence of the evolutionary algorithm is relatively poor, and the quality of the optimal solution is low. There have two ways to shorten algorithm's time efficiency, design parallel algorithm and create efficient algorithm which can reach convergence in fewer generations. Our research focus encompasses all of the above.

In this paper, we propose multi-objective heuristics based on consensus and membrane structure, called MOEA-MC, to infer phylogeny with the principles of parsimony and likelihood. And our work takes emphasis on achieving parallelism and convergence simultaneously, which parallelized by deploying on Spark, achieve fine convergence by adding consensus into each subpopulation in evolutionary algorithm. Additionally, to ensure each work node is assigned equal number of trees, we recommend using membrane structure to limit the number of trees in each subpopulation. Membrane structure can also restricted communication frequency between phylogenetic trees under different weights. We have compared MOEA-MC with other biological methods on three nucleotide datasets, and performed multi-objective assessment of biological properties by using several quality indicators and statistical tests. Finally, the rationality of the algorithm design will be verified by comparison with other methods in the literature. The main contributions of this work can be summarized as follows:

1) To develop effective parallel designs, we analyze the working process of multi-objective evolutionary algorithms by identifying computationally intensive operations that do not require global information.
2) A discussion on the main factors that slow down the convergence of that algorithm. We combining the consensus to maintain the topology and achieve accelerated convergence. In addition, a membrane structure is added to each working node to ensure the equal solutions under different weight and control the communication frequency between parallel sub-nodes.

The rest of this paper is arranged as follows: Section 2 introduced the materials and methods involved in MOEA-MC. In Section 3, we analyzed how to combine consensus and membrane structures in multi-objective evolutionary algorithm, and showed the pseudo code of MOEA-MC. The related process of parallel MOEA-MC is present in Section 4. The experimental results are discussed in Section 5. Finally, Section 6 summarizes our work and outlines future work.

II. RELATED WORKS

In this section, we depict the intuition and technical details of phylogenetic reconstruction, discuss the reasons why reconstruction phylogenetic development reveals the NP-hard nature [2], [13], and explore how to solve this NP-hard problem [14].

The diversity of creatures in nature reflects the diversity of evolutionary patterns, leading to different representations of species. How to explain this evolutionary process is the goal of evolutionary biologists. Analysis of biomolecular data can account for mutations and replacement events observed at the nucleotide level, which are the source of evolutionary diversity. In phylogenetic analysis, an $N \times M$ aligned molecular sequence (N is the number of organism and each one contains M features or sites) is processed to reconstruct the hypothesis of evolutionary events related to this sequence. The evolutionary relationship is modeled by inferring the system tree $N \times M$ where branch set E specifies the ancestor relationship between the organisms in node set V. In evolutionary biology, the leaf nodes of a phylogenetic tree are species, or biomolecular sequences or biological entities, but in this paper, the leaf nodes of the phylogenetic tree are all biomolecular sequences, such as gene sequences or protein sequences. Moreover, taxa on the leaf nodes are collectively named Operational Taxonomic Units (OTUs). Correspondingly, the internal node is called Hypothetical Taxonomic Units(HTUs), which represents the possible ancestors of the leaf nodes [15]. The relative distance between objects represents the evolutionary closeness between the objects. The longer the branch length, the more likely it is to mutate.

Accordingly, it can be concluded that the purpose of phylogenetic tree reconstruction is to find the phylogeny $T = (V, E)$ that meets certain biological quality standards. In evolutionary biology, the leaf nodes of a phylogenetic tree can be species, biomolecular sequences or biological entities, but biological molecular sequences (such as gene sequences or protein sequences) are used herein. The leaf nodes on the evolutionary tree are biological objects, the length of branch indicate the kinship distance among leaves and the topology of the evolutionary tree describes the evolutionary relationship of these objects. Evolutionary tree can be divided into rooted tree and unrooted tree according to whether it can represent the evolutionary order. The root of rooted tree is the closest common ancestor of all leaf nodes and the direction of evolution is from root to leaf. The unrooted tree has no root node and cannot represent the evolutionary order between nodes. Reconstruct the possible evolution tree according to a sequence file with n objects, the number of unrooted tree $\mathrm{U}(n)$ and rooted tree $\mathrm{R}(n)$ can be computed as follows [16]:

$$\mathrm{U}(n) = 1 \times 3 \times 5 \times \cdots \times (2n-5) = (2n-5)!! \quad (1)$$

$$\mathrm{R}(n) = 1 \times 3 \times 5 \times \cdots \times (2n-3) = (2\mathrm{n}-3)!! \quad (2)$$

As the number of species grows, the reconstruction of phylogenetic trees (whether rooted or unrooted) has become an NP-hard problem. For example, given a sequence with 50 objects, we can get 2.84×10^{74} unrooted trees and 2.75×10^{76} rooted trees.

A. OBJECTIVE FUNCTION

Reconstruction can be basically divided into four steps. Firstly, get the biomolecular sequence. Thanks to the development of sequencing technology, this can be obtained from major gene banks or biological information databases, such as GenBank, European Molecular Biology Laboratory (EMBL). Secondly, perform data preprocessing such as site alignment. Thirdly, choose one evolutionary reconstruction model which has already emerged in biological, namely the speculation or hypothesis of the evolutionary laws of species. Finally, reconstructed phylogenetic tree by an algorithm based on the evolutionary reconstruction model. Originally, we list all possible evolutionary trees according to the given sequence, and then recommend the best one. With the rapid development of bioinformatics and larger reconstruction sequences, it is inadvisable yet to enumerate all possible trees. Under the optimal standard requirements, the huge amount of computation leads search mechanism to use heuristic technology [17], which can find the appropriate solution for large or even datasets within reasonable runtime. Of course, for affordable small datasets, we can still consider using exhaustive or precise search techniques. The methods to finish the fourth step can be divided into: based on optimal principle and based on no-optimal principle. The former reconstructs a tree with comparable evaluation values, so the best tree can be found. The latter obtains a phylogenetic tree based on algorithmic steps and cannot be compared. Maximum Parsimony Method(MP) [18], [19] and Maximum Likelihood Method (ML) [20] are the two most classic algorithms based on the optimal principle. The latter category usually classified as distance-based methods which uses the difference of the sequence to construct the distance matrix, and then reconstructs the evolution tree, such as neighbor joining (NJ) [21],

and Bayesian Inference (BI). Because the former have exact comparable values, most multi-objective optimization methods infer phylogeny by maximizing parsimony and likelihood in literature. In this paper, we consider phylogenetic reconstruction as a dual objective optimization problem involving two widely used biological objective functions: parsimony and likelihood, as reported in the literature. And the parsimony value is obtained using Fitch's algorithm [22], the likelihood score is calculated using the Felsenstein algorithm [23].

1) MAXIMUM PARSIMONY

Using maximum parsimony method to reconstruct phylogenetic trees is first proposed by Camin(1965) [24] and Hein(1990,1993) [25]. The principle of the maximum parsimony method is based on the Ockham's razor, which is a philosophical statement that tends to choose simpler than a complex competitive process. In other words, maximum parsimony method follows the principle of minimal change, that is, the fewer mutations or replacement events required for the evolutionary process, the closer to the fact. Given a dataset that have n aligned sequences and each sequence has m features, we can inferring a tree $T = (V, E)$. The parsimony calculation needs to set the ancestor sequence of each node in advance, which can be solved by adopting the bottom-up approach [22]. After assigning the ancestor sequence, the calculation formula for the parsimony score $P(T)$ of the tree T is defined as [26]:

$$P(T) = \sum_{i=1}^{m} \sum_{(u,v)\in E} C_i(u, v) \tag{3}$$

where $u, v \in V$ and there have branch $(u, v) \in E$ to link them, $C_i(u, v)$ is an integer value used to quantify the observed mutation events between u and v, and C is the cost matrix, like $C_i(u, v)$ indicates the difference between u and v at the i site, and calculated as follows:

$$C_i(u, v) = \begin{cases} 1, & if\ u_i \neq v_i, \\ 0, & otherwise. \end{cases} \tag{4}$$

u_i and v_i are the sequence state values of the ith character of u and v, After getting the maximum parsimony value of the branch, the next step is to calculate the maximum parsimony value of tree.

2) MAXIMUM LIKELIHOOD

The entry point of the maximum likelihood method is the branch length of the phylogenetic tree. There is a positive correlation between the length of the branch of the phylogenetic tree and the evolution time between the leaf nodes. And it is obvious that evolution time is closely related to the probability of variation. The maximum likelihood method was originally used to obtain parameters of probability models in statistics. Joseph Felsenstein (1980) first proposed the application of the maximum likelihood method to phylogenetic inference. In phylogeny, for a series of phylogenetic trees reconstructed from a given sequence, the one with the largest likelihood value is the closest to the real phylogenetic tree. Therefore, the likelihood-based phylogenetic tree reconstruction scheme first reconstructs the possible phylogenetic trees, calculates the likelihood values of each phylogenetic tree one by one, and finally considers the phylogenetic tree with the largest likelihood as the optimal. Let D be a collection of n aligned sequences with N characters per sequence (characters can concluded as $\Sigma = \{A, C, G, T\}$). M is an evolutionary model used to describe evolutionary hypotheses, which provides a mutation probability at the nucleotide level and determined the ancestral sequences in advance(such as JC69 [27], HKY85 [28], GTR [29], TN93 [30], K80 [31]). The phylogenetic topology $T = (V, E)$ is a description of the evolutionary hypothesis. The likelihood of T can be calculated as:

$$L(T) = P(D|T, M) = \prod_{j=1}^{N} L_j(T) \tag{5}$$

where $L_j(T) = P(D_j|T, M)$ is the likelihood at character state j and the detailed formula is:

$$L_j(T) = \sum\nolimits_{r_j} C_j(r_j, r) \cdot \pi_{r_j} \tag{6}$$

where π_{r_j} represents the stationary probability for the state $r \in V$ appears, when character state r_j is defined from alphabet Σ. And $C_j(r_j, r)$ is the partial conditional likelihood at site j with rooted at node r, and $r_j \in \Sigma$ represents all possible state at site j. Let $r \in V$ be a HTU which have descendants u and v, then the calculation for $C_j(r_j, r)$ is:

$$C_j(r_j, r) = \left(\sum\nolimits_{u_j} C_j(u_j, u) \cdot P(r_j, u_j, t_{ru})\right) \cdot \left(\sum\nolimits_{v_j} C_j(v_j, v) \cdot P(r_j, v_j, t_{rv})\right) \tag{7}$$

where u_j and v_j represents character state of the node u and v at site j. t_{ru} and t_{rv} are the branch lengths of connecting node u and v to the node r respectively which are given by $(r, u) \in E$ and $(r, v) \in E$. $P(r_j, u_j, t_{ru})$ indicate the probability of transfer r_j of the node r to u_j of the node u during the evolution time t_{ru}, and $P(r_j, v_j, t_{rv})$ have the same definition. In addition, the value of $P(r_j, u_j, t_{ru})$ and $P(r_j, v_j, t_{rv})$ are all provided by the evolution model M.

B. MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM

A single optimization problem considers only the maximization (or minimization) of an objective function. Differently, multi-objective optimization problems involve multiple targets, and usually conflict with each other. The application of multi-objective optimization in phylogeny represents a hopeful solution to deal with main source of inconsistency that may affect the reliability of phylogenetic reasoning. According to [32], the study of phylogenetic reconstruction can be divided into two aspects. On the one hand, a series of multi-objective evolutionary algorithms have been successfully proposed to solve conflict information in different

data sets. On the other hand, other studies focus on solving inconsistencies caused by phylogenetic analysis using different optimal criteria. The most controversial of these is the conflict between parsimony and likelihood. Studies [33], [34] have shown that these two standards may lead to conflicting evolutionary hypotheses.

Hence the need to address potential conflict between different optimal criteria [34], which turn into the main source of inconsistency in phylogenetic research. A way to address this issue involves introduce a multi-objective formulation of the problem. In real world, it is often encountered problems are usually composed of multiple goals or several evaluation indexes that conflict and affect each other. While optimization target exceed one and need meet them simultaneously, called it as multi-objective optimization problem(MOP) [7], [8], can be formulated as follows:

$$\begin{aligned} &maximise\ F(x) = (f_1(x), \ldots, f_m(x))^T \\ &subject\ \text{to}\ x \in \Omega. \end{aligned} \tag{8}$$

where Ω is the search domain, x is the decision variable, m indicate the number of objective functions, and $F : \Omega \rightarrow R^m$, R^m denote the solution space [35]. When $m = 1$, the optimization problem is single-objective optimization problem, if $m \geq 2$ called it as multi-objective optimization problem. In general, there are multiple objectives or evaluation criteria for MOP, and each target is mutually constrained. While optimizing one goal, it is at the cost of reducing the performance of other targets. Generally, the multi-objective optimization problem does not have a single optimal solution, but a set of approximate optimal compromise solutions. The traditional optimization algorithm can only obtain a compromise solution in one operation, so the solution efficiency for multi-objective optimization problems is too low to meet the actual application requirements. The evolutionary algorithm(EA) takes the population as the evolution unit which can obtain a set of approximate optimal solutions in one effective iteration [36]. Multiple individuals in EA evolved at the same time, which can reduce the importance of individual that result in reduce the probability of falling into the local optimal ''trap'' [37]. At present, many multi-objective evolutionary algorithms have been proposed, such as representative dominance-based approach NSGA-II [11] and decomposition-based MOEA/D [12] and PhyloMOEA [38]. These classic algorithms were performed significant in this field and usually acted as reference when proposed new work to solve MOP [39].

In multi-objective optimization, there is usually no viable solution that can minimize all objective functions at the same time. In other words, there is no way to improve the solution in any target without lowering any other goals. Therefore, our goal is to search for the Pareto optimal solutions which one have no other solution can dominate it in all objectives. For example, f with different suffixes represents different maximization functions, $x_1, x_2 \in \Omega$, a feasible solution x_1 is dominated by x_2, if:

1. $f_i(x_1) \leq f_i(x_2)$ for all functions $i = \{1, 2, \ldots, m\}$
2. $f_j(x_1) < f_j(x_2)$ for at least one objective $j = \{1, 2, \ldots, m\}$

The points in the objective space corresponding to the Pareto-optimal are non-dominated, and all of them formed Pareto-frontier.

C. CONSENSUS

The concept of consensus have been mentioned in [40], which has introduced that consensus tree can summarizes the topological features of multiple trees and integrates them into single tree. The consensus tree can be divided into several categories (such as strict consensus tree, majority rule consensus tree, loose consensus tree, and greedy consensus tree) [40] according to the integration method. MOEA-RC [41] using the majority rule consensus to retain branch features during evolution, which have certificated consensus can help MOEAs converge in less generations.

Our paper is also picked the majority rule consensus. As MOEA/D [12] depicted that neighbors are likely to have similar search directions. So the number of solutions required to calculate consensus should be greater than 2. In addition, if select all solutions to calculate consensus, the results will be completely homogeneous. It also can result in few elites in the solution and lose the correct consensus. In summary, we chose the suitable number: 3, which can reduce calculation and ensure the reliability of the consensus. In our work, consensus can accelerate convergence when act on crossover and mutation. The consensus branches under different weights are considered as correct branches in the current population, so evolutionary algorithm will protect the topology of consensus in crossover and mutation. This retention can reduce the overall execution of evolutionary algorithm and also speed up searching operation.

D. MEMBRANE STRUCTURE

In 2004, Zhang [12] proposed a multi-objective evolution algorithm MOEA/D based on decomposition. However, Zhang [42] found that the Pareto front lacked diversity. Take researches on MOEA/D found that some (not all) solutions are selected among sub-problems, and there may be many sub-problems corresponding to the same non-dominated solution, which leads to the loss of solution diversity. In order to solve this problem, Zhang [42] designed a multi-objective evolutionary algorithm combining membrane structure to reduce the number of sub-problems and improve the probability that each sub-problem has different solution. In biology, membrane plays a vital role in the structure and function of living cells. Membrane structure can help ensure that a sub-problem will have multiple solutions, where the membrane structure refers to the structure of the membrane computing model. Membrane computing is a branch of natural computing. It is a computational model that is inspired by the structure and function of cells and tissues or organs composed of cells. In the ten years since the concept of ''membrane computing'' was put forward, the computing theory, models,

algorithms, and applications of membrane computing have developed rapidly. Membrane computing provides new distributed parallel information processing methods and technologies for computer science, promotes the development of new high-performance computing technologies, and provides a new way to solve computationally difficult problems.

Evolutionary evolution within the membrane eliminates solutions with the worst performance. Therefore, in a sub, the best solution to choose is relatively more. Through multiple iterations, each membrane structure solution is considered to be the best solution to the sub-problems of the membrane structure. Conversely, evolutionary algorithm hold potential capability to be parallelized which have been designed as parallel genetic algorithms(PGAs) [43]. The membrane structure can well complete the evolution inside, and divide all the current individuals into multiple subpopulations. Through the evolution of the subpopulation in the membrane structure, the local optimal solution and the exchange between adjacent membrane structures are used to seek the global optimal solution. Membrane structure can divide the population into specified sizes. Similar to the biological membrane structure, by defining a closed space, the interior can maintain a different biochemical environment than the outside world. Each subgroup is regarded as an cell with unique membrane which can restrict the account of trees in one 'cell' and limit the timing when to exchange maximum, minimum and updates optimal solutions. The specific implementation steps are:

1) Initialization: Divided the object space into multiple membrane structures and the solution for each membrane structure is initialized.
2) Each subpopulation is independent and concurrent, to completes genetic manipulation and evaluates individuals. Determine whether the iteration meets the exchange requirement by the timer which is set by the membrane structure. If reach, replace the worst solution with the excellent solution in the neighbor subgroup through membrane.
3) Iterate through the second and third steps until the appropriate individual is found or the specified number of iterations is completed.

III. MOEA-MC

After above detailing depiction the superiority about consensus and membrane structure, we designed a novel MOEA which integrate membrane structure and consensus. Lemmon [44] have concluded that four trees can generate the optimal consensus. Therefore, we apply every membrane divided into four subpopulations directly, and the consensus is calculated from the optimal solution of the four parts. Each subpopulation develops independently which has own development direction and consensus. Thus they evolves alone with protect consensus through the genetic operators of evolutionary algorithm. The independence of membrane structure, which are suit to decompose, lead us to employ the weighted sum method [45] and decomposed the multi-objective optimization problem into multiple single-objective optimization problems by their weight [46]. Thus, each membrane corresponds to a weight vector. With the previous uniform setting of weight vector, the better distribution of the final non-dominated solution set. Based on the above analysis, we adapted the MOEA/D algorithm by integrating consensus and membrane structure to tackle the phylogenetic inference problem. Algorithm 1 shows the pseudo-code of the MOEA-MC, where D corresponds to a sequence-aligned biomolecule file in PHYLIP format, m is the number of membrane structure, mp and mo are the mutation rates and mutation operator respectively, pc is the probability of perform crossover, ei is the exchange interval and It corresponds to the number of search iterations which are pre-set. The following subsections describe details of the algorithm.

Algorithm 1 MOEA-MC Pseudo Code

1. **Input: D**, m, **S**, It, mo, pc, ei
2. **Output:** A $\boldsymbol{P}$ population of trees (non-dominated solutions found by the algorithm).
3. Phylogenetic trees $T \leftarrow$ *initialize* ($\boldsymbol{D}$, $\boldsymbol{N}$, $\boldsymbol{4}$, $\boldsymbol{S}$)
4. Generate weight vector W_m with well distributed.
5. **while** stop condition is not reached **do**
6. for each tree $tr \in T$ do
7. MP[tr] ← Fitch's algorithm
8. ML[tr] ← Felsenstein's algorithm
9. Membrane[N] ← Redistribution
10. Consensus[N] ← the majority rule consensus
11. **for each** $p \in P$ **do**
12. [tr_1, tr_2] ← *binary_tournament_selection(p)*
13. P[p] ← *crossover*(tr_1, tr_2, pc)
14. P[p] ← *mutation*(T, mo, mp)
15. P[p] ← *exchange*(T, ei)
16. **end while**
17. **Return** $\boldsymbol{P}$

1) Initialization:
Transform file D into N * 4 * S phylogenetic trees by using a rearrangement method, and generates a well distributed weight vector $W_m = \{w_1, w_2, \ldots, w_m\}$.

2) Calculate MP and ML:
Different objective functions have different values. In order to better measure the pros and cons of the solution on the objective function, each value is standardized as follows.

$$\bar{f}_i = \frac{f_i - z_i^*}{z_{nad}^i - z_i^*} \tag{9}$$

$\bar{f}_i$ is the normalized result of the i-th objective function among the m objective functions, $z^* = (z_1^*, \ldots, z_m^*)$ and $z^{nad} = (z_1^{nad}, \ldots, z_m^{nad})$ are the optimal and worst of the m objective functions.

3) Redistribution and Calculate consensus:
Then calculate the fitness value of each solution according to the following formula.

$$G^{ws}(x|w_i) = \sum_{j=1}^{n} w_i^j f_j \tag{10}$$

where $G^{ws}(x|w_i)$ is the fitness value for solution x under the weight w_i, f_j is the value of the jth objective function. Its value is the sum of the product of the weight and the corresponding value of each dimension of the objective. And call equation 10 as the weighted sum method which decomposing multi-objective optimization problems into n subs which correspond to w. The population is divided into several subs, and the trees in each sub-population are sorted according to fitness value. According to previous definition, we can computed N the majority consensus and broadcasted to each working node later.

4) Generate descendants:

Take binary_tournament_selection on subpopulations to ensure each one have two phylogenetic trees. Perform crossover and mutation on them and generated descendants.

5) Selection:

Merge the parent and child. Sort them inside of membrane and eliminate half of the phylogenetic tree with low fitness value.

6) Exchange:

Judges whether reach the migration conditions ei. If iteration intervals have arrived, take the migration operation: replace the four optimal solutions on the adjacent with the eight worst solutions on the target. Otherwise, pass.

7) Stop or continue:

Determine if the stop condition is met. If it is satisfied that stop algorithm, otherwise returns to step2 and continue the execution.

IV. PARALLEL DESIGN

At present, solving the computationally demanding optimization problems in bioinformatics mainly relies on the combination of biological heuristic algorithms and parallelism. After detailing the main features of genetic algorithm MOEA-MC that can effectively overcome the premature convergence problem of standard genetic algorithm and has strong global search ability, we need to design a reasonable and efficient parallel frame which fit in implement MOEA-MC. In this sense, using parallel platforms or parallel development kits [47] allows us to take advantage of the division of labor and high-speed communication to leverage this architecture in an efficient manner. At present, we have the popular parallel platforms such as OpenMP, MPI [48], Hadoop [49]–[52] and Spark [53]–[56]. With the rapid development of computer technology, the coordination between the subtasks of parallel algorithms has been undertaken by third-party programs. Developers just need to note the parallel mechanism, instead of how to coordinate the work of the cluster. These third-party programs are usually presented in the form of development kits or in the form of a platform. Compared with parallel implementations based on development kits [47], platform such as Hadoop and Spark is more simple to implement and more scalable.

Spark [57] which developed by AMP Labs at the University of California at Berkeley have outstanding features such as high availability, high processing speed and fault tolerance. First, Spark uses an efficient DAG execution engine that can quickly process data streams based on memory. Second, Spark has strong fusibility and can be easily integrated with other technologies. Spark also has its own resource manager and schedulers, such as standalone mode which implements a built-in resource manager and scheduling framework. In addition, compared to the temporary files in Hadoop's local hard disk storage process, Spark uses memory as a temporary storage have greatly speed up the data processing capabilities. Therefore, Spark's parallel and iterative structure is very suitable for information mining of biological data and can confirm to parallel and improve MOEA-MC. Therefore, we will design a parallel algorithm based on Spark, because this combination represents one of the most effective choices for dividing the computer CPU core into multiple working nodes and performing time-consuming objective function calculations in parallel. Follows is the modified and parallelized MOEA-MC.

In order to develop an efficient parallel approach, the first step we must perform is to identify operations that do not require global information. The initialization operation requires the entire sequence file information which is not suitable to parallel. Calculate fitness value can be parallelized because the calculation of likelihood and parsimony do not show a dependency between the phylogenetic trees. Consensus is also only related to the trees inside the membrane structure, so it can be operated in parallel. Generate descendants need parents and the corresponding consensus which not related to other working nodes, so can be carried out in parallel. Merge child and the parent into entirety absolutely can be directly executed by the shuffle operation in Spark. Determining whether to exchange the optimal solution is depends on the iteration interval designed by membrane structure which can also control the information diffusion between subgroups. After theoretical analysis, the most time-consuming operations in the MOEA-MC can be executed in parallel. Next, using the Spark parallel structure, we can use 'parallelize' in Spark to create RDD which is parallel data corresponding to each step and can be used to set parallel processing operation. In summary, parallelized MOEA-MC is a parallel algorithm which fit in deploying on Spark.

V. EXPERIMENTS AND ANALYSIS

In this section, we conducted a series of experiments to evaluate the performance of parallel MOEA-MC. In addition, we also presented and analyzed the experimental results of MOEA-MC on parallel performance and biological quality.

A. CONFIGURATION

For experimentation purposes, we have used three real-world biological datasets whose details of the sequences and their corresponding sources have been showed in TABLE1. Our experimental platform is one PowerEdge R730 computer with 2.40GHz (32 core) and operating on Ubuntu 5.4.0-6. General Time Reversible evolutionary model (GTR) is used

TABLE 1. Real nucledotide datasets.

Data	Sequence size	Nucleotide size	Description
rbcl_55	55	1314	rbcL plastid gene
mtDNA_186	186	16,608	Human mitochondrial DNA
ZILLA_500	500	759	500 rbcL sequences from plant plastids

TABLE 2. Common algoriyhm configurations.

Parameter	Value
Population size	100
Number of iteration	100
Selection operator	Binary tournament[28]
Hybrid operator	Prune-Delete-Graft [29]
Hybrid probability	0.8
Mutation probability	0.2
Mutation operator	NNI
Evolutionary model	GTR[30]

to implement the ancestor sequence in advance. In addition, experimental comparison of various parameter variables of the evolutionary algorithm to find out what input parameter configuration can better improve the quality. TABLE2 lists the common algorithm configurations. And the aggregation function used by MOEA/D in our work is Tchebycheff.

B. PARALLEL PERFORMANCE

First, we have executed MOEA-MC at different parallelism to observe the relationship between the execution time and parallelism. Fig1 shows the runtime of MOEA-MC with 100 iterations on *rbcL_55* dataset. By experimenting with the increase and decrease of the degree of parallelism of MOEA-MC on three data sets, we found that the most suitable parallelism is different on different datasets. On ZILLA_500, the optimal degree is achieved when the degree of parallelism is preset as 24, and the best performance in mtDNA_186 was achieved at 32. Therefore, MOEA-MC can get less time with the appropriate parallelism which have proved the effect after deployed MOEA-MC on Spark, and with the parallelism increases that MOEA-MC's execution time gradually decreases until reach its balance.

MOEA-MC was designed to resolve tree reconstruction, so we need take comparison to judge if MOEA-MC can

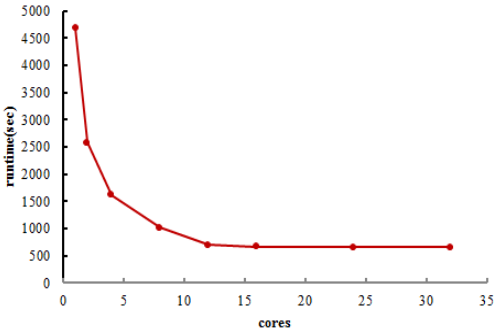


FIGURE 1. Evolution of runtime for MOEA-MC performed on *rbcL_55* dataset.

improve the objective values. In order to testify the performance, we tested the maximum parsimony and maximum likelihood of MOEA-MC in three real-world datasets and compared with several classic multi-objective evolutionary algorithms (MOEA/D [12], NSGA-II [11] and PhyloMOEA [38]). Take experiments on MO-Phylogenetics [59] (which is a tool to infer phylogenetic trees) to got the final maximum parsimony and maximum likelihood value of MOEA/D [12], NSGA-II [11] and PhyloMOEA [38]. In TABLE3, we reports comparisons of the maximum likelihood (ML) with the reference several multi-objective algorithms (MOEA/D [12], NSGA-II [11] and PhyloMOEA [38]). These maximum likelihood values are all multiplied by -1 in order to make goal become research the minimum of two functions in uniform standard. The other objective MP experiment is show in Table4. Incidentally, the value in each table is all take the best during all iterations. The purpose of MOEA-MC algorithm is to decompose the reconstruction task into multiple workers and calculate the objective function in parallel. Farther, reserved consensus to speed up evolution, and set the membrane structure to ensure that the number of solutions in the working node is not out of balance. But these settings can't achieve more earnings since the various more complex calculations and MOEA-MC in the machine is still running in serial mode. The meaningless results are showed in TABLE3 and TABLE4, which have

TABLE 3. The maximum likelihood score for MOEA/D, NSGA-II, PhyloMOEA and MOEA-MC.

Algorithm	*rbcl_55*	*mtDNA_186*	*ZILLA_500*
MOEA/D	22,169.6	39,937.5	84,715.9
NSGA-II	22,193.3	39,942.5	84,719.4
PhyloMOEA	22,200.1	39,938.2	84,704.6
MOEA-MC	22,155.0	39945	84,744
MOEA-MC (parallelized)	**22,145**	**39,931.95**	**84,674**

TABLE 4. Maximum parsimony score for MOEA/D, NSGA-II, PhyloMOEA and MOEA-MC.

Algorithm	*rbcl_55*	*mtDNA_186*	*ZILLA_500*
MOEA/D	4979	2461	17,194
NSGA-II	4979	2463	17,192
PhyloMOEA	4982	2461	17,191
MOEA-MC	4980	2462	17,192
MOEA-MC (parallelized)	**4978**	**2459**	**17,164**

certified that only add consensus and membrane can't achieve better performance but parallel design can change this mode. This can also be understood as the reason why parallel algorithms are getting more and more attention on multi-objective problems.

TABLE 5. Runtime (seconds) for MOEA/D, NSGA-II, PhyloMOEA and MOEA-MC.

Algorithm	*rbcl_55*	*mtDNA_186*	*ZILLA_500*
MOEA/D	1302.64	65,620.20	13,082.54
NSGA-II	1287.07	66,010.94	12,812.62
PhyloMOEA	2163.40	>24 h	>24 h
MOEA-MC	4717.0	>24h	44,640
MOEA-MC (parallelized)	**840(32)**	**14760(32)**	**6480(24)**

The design concept of the MOEA-MC is to achieve the purpose of shorten runtime by using modern multi-core cluster technology. In the TABLE5, it have represents that MOEA/D [12], NSGA-II [11], PhyloMOEA [38] and MOEA-MC run at different datasets have showed a great different execution time. The most outstanding results have been highlighted in bold. Farther, we have annotated the parallelism of MOEA-MC, it have better performance with 24 cores in ZILLA_500 rather than 32 cores in other datasets. Obviously, MOEA-MC is bold in all data sets. Basically, MOEA-MC's execution time is reduced by 50% compared to other classic algorithms. Table 3-5 have also shows the performance of non-parallel MOEA-MC in MP, ML and runtime. It is obvious that the overall performance of the non-parallel MOEA-MC is slightly inferior, even if the ML value obtained in rbcl_55 is less than NSGA-II and MOEA/D, and got the similar output with them on MP. It can be concluded that non-parallel MOEA-MC is worse than parallelized MOEA-MC on all datasets. It is worth mentioned that we have taken all experiments under same environment, and picked the best one as final.

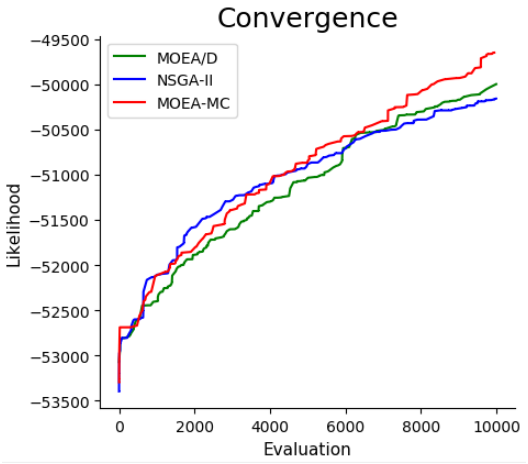


FIGURE 2. Convergence for MOEA-MC performed on *mtDNA_186*.

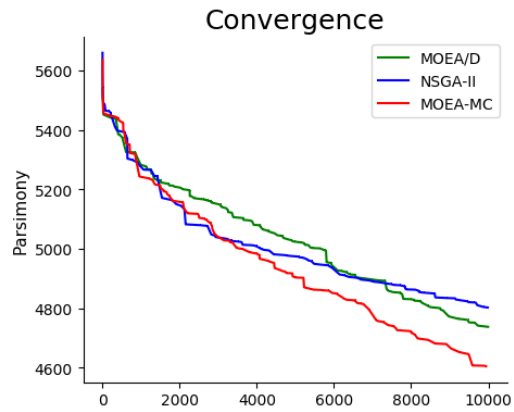


FIGURE 3. Convergence for MOEA-MC performed on *mtDNA_186*.

In order to assess that combine consensus and MOEA can improve convergence like [41], we include a comparison with other approaches from literature. Fig 2-5 have clarified that MOEA-MC can achieve better convergence. Figures 2 and 3 have showed the changes of MP and ML on the *mtDNA_186* dataset as the iteration progresses. It can be found that the convergence performance of MOEA-MC on ML is better than others, and always been in a dominant position during the iteration process. Although the convergence performance on MP is slightly worse than NSGA-II, it can still maintain the NSGA-II after the iteration on. Figures 4 and 5 show the MP and ML changes of the three algorithms on the *rbcl_55* dataset as the iteration progresses. It can be found that the convergence performance of MOEA-MC is better than the other algorithms. Although the convergence speed at the beginning is slightly worse than NSGA-II, MOEA-MC can achieve convergence earlier than NSGA-II.

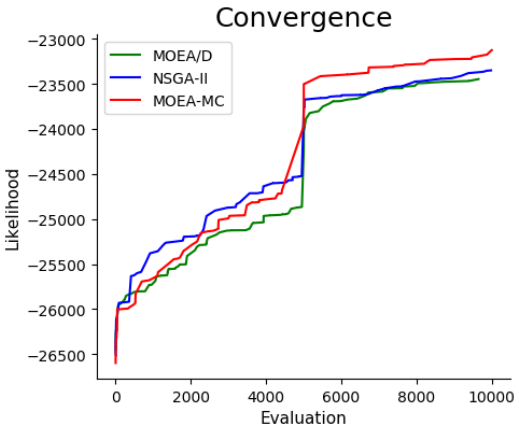


FIGURE 4. Convergence for MOEA-MC performed on *rbcl_55.*

After assessing biological performance, we now focus on verifying the multi-objective performance of the inferred solutions. The main purpose of this section is to check whether the use of hybrid parallel design results in poor quality of multi-objective solutions. In order to evaluate multi-objective performance, we used the widely used Pareto front indicator. As depicted in section 2.2, there is no optimal solution for multi-objective problems, the goal of multi-objective evolutionary algorithm is to find all feasible solutions in the search space, and then find solutions which are not dominated by another solutions. We can got the optimal Pareto solutions that have no other solutions is better than them through multi-objective evolutionary algorithms.

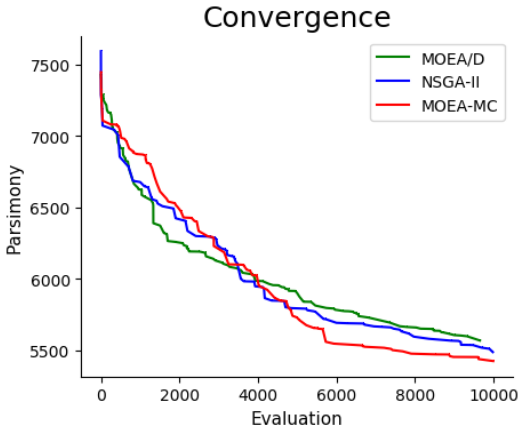


FIGURE 5. Convergence for MOEA-MC performed on *rbcl_55.*

In order to verify MOEA-MC not only have power to speed up in parallel but also have superiority in search, we also made experiments about the Pareto frontier. The Pareto frontier of MOEA-MC and NSGA-II, MOEA/D on three datasets is shown in Figure3 to 5, in which we have got results via 100 iterations. Evidently, the results show that MOEA-MC is locate at upper left which remain that MOEA-MC got the lower parsimony and higher likelihood meanwhile. The coverage relationship demonstrates that solutions obtained

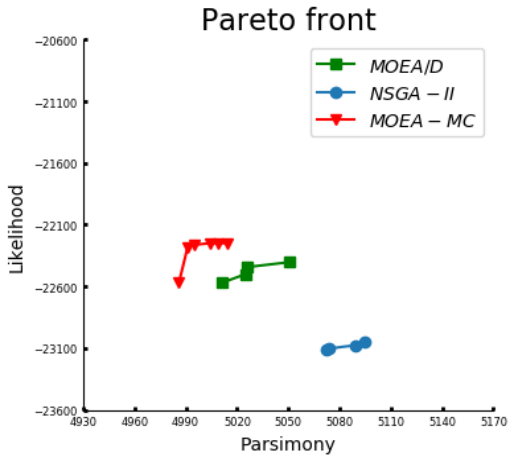


FIGURE 6. Pareto fronts generated from MOEA-MC, NSGA-II and MOEA/D over the *rbcl_55* dataset.

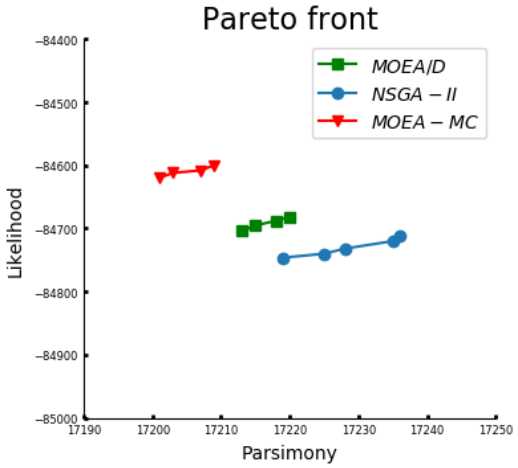


FIGURE 7. Pareto fronts generated from MOEA-MC, NSGA-II and MOEA/D over the *ZILLA_500* dataset.

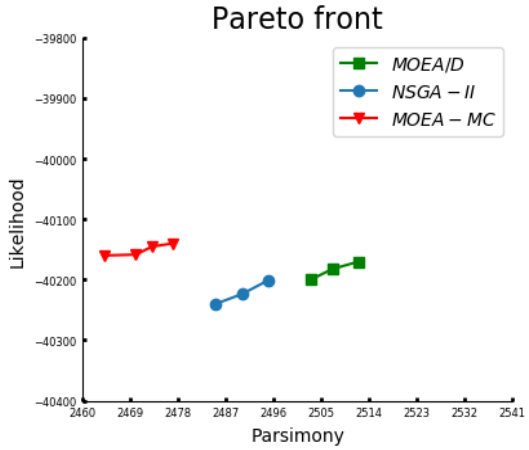


FIGURE 8. Pareto fronts generated from MOEA-MC, NSGA-II and MOEA/D over the *mtDNA_186* dataset.

by MOEA-MC are non-dominated by solutions from other methods in all the data sets. In this case, MOEA-MC and NSGA-II, MOEA/D also have non-dominated solutions. Also, MOEA-MC is the method that most contributes to the

global Pareto-frontier for all data sets. Until now, MOEA-MC has proved it can reduce runtime by parallel and got smaller ML and MP values by unique algorithm components. Also, MOEA-MC got non-dominated Pareto fronts with only 100 iterations. Most importantly, our proposed MOEA-MC outperform on all indicators, indicating apply parallel processing for multi-objective evolutionary algorithms, which can achieve faster, more accurate to referring phylogeny history.

VI. CONCLUSION AND FUTURE LINES

In this paper, we have proposed parallelized multi-objective evolutionary algorithm based on consensus and membrane structure (MOEA-MC). Consensus in each subpopulations can reserve the best topologies that resulted evolutionary algorithm get converged in shorter runtime. By studying the independence between the sub-populations in each time-consuming step of the evolutionary algorithm, the steps without global information can be designed to be executed in parallel, which can fundamentally reduce the execution time. In order to eliminate the imbalance between parallel working nodes, we have used membrane structure to control the solutions number under different weights. In parallel design section, a comparative analysis was carried out between the existing parallel approaches. With the design of the parallel algorithm, MOEA-MC has chosen Spark as parallel tool. Parallelized MOEA-MC also can control the communication frequency between each work node by setting migration interval. With the standalone cluster mode of the Spark, the degree of parallelism is controlled with set CPU cores. Speedup analysis on different system sizes allows us to determine the main factors controlling parallel performance and the appropriate parallelism for different data sets.

Moreover, the analysis of multi-objective results has pointed out that MOEA-MC preserves the search capabilities of the original evolutionary algorithm, giving rise to high-quality sets of Pareto solutions in reduced execution time. By locating the Pareto optimal solution obtained in 100 iterations in the objective function graph, it is obvious that the Pareto front obtained by MOEA-MC can dominate other solutions. In conclusion, our research shows that applying parallel methods can better cope with this huge computing challenge.

Although the results shown in this work are promising, there still are important issues to improve in the algorithm see, e.g. [60]–[63]. As future work lines, we aim to study new parallel approaches such as machine learning [64]–[66] and deep learning [67]–[69] for phylogeny. We will address the development of asynchronous algorithms for pure shared memory environments involving a large number of processing cores. And reduce the shuffle operation in Spark as much as possible.

ACKNOWLEDGMENT

(Qianqian Zhang and Jun Zhang contributed equally to this work.)

REFERENCES

[1] Z. Guiwei, Z. Jue, G. Wei, and L. Qiong, "Bioinformatics analysis of the AT-hook gene family in rice," *Chin. Bull. Botany*, vol. 49, no. 1, p. 49, 2014.

[2] S. K. Pal, S. Bandyopadhyay, and S. S. Ray, "Evolutionary computation in bioinformatics: A review," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 5, pp. 601–615, Sep. 2006.

[3] X. Xia, "Phylogenetic relationship among horseshoe crab species: Effect of substitution models on phylogenetic analyses," *Syst. Biol.*, vol. 49, no. 1, pp. 87–100, Jan. 2000.

[4] A. Rokas, B. L. Williams, N. King, and S. B. Carroll, "Genome-scale approaches to resolving incongruence in molecular phylogenies," *Nature*, vol. 425, no. 6960, pp. 798–804, Oct. 2003.

[5] J. Handl, D. B. Kell, and J. Knowles, "Multiobjective optimization in bioinformatics and computational biology," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 4, no. 2, pp. 279–292, Apr. 2007.

[6] M. Villalobos-Cid, M. Dorn, R. Ligabue-Braun, and M. Inostroza-Ponta, "A memetic algorithm based on an NSGA–II scheme for phylogenetic tree inference," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 776–787, Oct. 2019.

[7] C. C. Coello, C. Dhaenens, and L. Jourdan, Eds., *Advances in Multi-Objective Nature Inspired Computing*, vol. 272. Springer, 2009.

[8] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering–based evolutionary algorithm for large–scale many–objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 97–112, Feb. 2018.

[9] T. Wang, H. Luo, W. Jia, A. Liu, and M. Xie, "MTES: An intelligent trust evaluation scheme in sensor–cloud enabled industrial Internet of Things," *IEEE Trans. Ind. Informat.*, to be published.

[10] P. Lemey, M. Salemi, and A.-M. Vandamme, Eds., *The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA–II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[12] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[13] B. Chor and T. Tuller, "Maximum likelihood of evolutionary trees is hard," in *Proc. Annu. Int. Conf. Res. Comput. Mol. Biol.* Berlin, Germany: Springer, 2005.

[14] T. Wang, L. Qiu, G. Xu, A. K. Sangaiah, and A. Liu, "Energy-efficient and trustworthy data collection protocol based on mobile fog computing in Internet of Things," *IEEE Trans. Ind. Informat.*, to be published.

[15] S. Santander-Jimenez and M. A. Vega-Rodriguez, "Parallel multiobjective metaheuristics for inferring phylogenies on multicore clusters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1678–1692, Jun. 2015.

[16] J. Felsenstein, "The number of evolutionary trees," *Syst. Zool.*, vol. 27, no. 1, p. 27, Mar. 1978.

[17] J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving* (Artificial Intelligence). 1984.

[18] J. De Laet, "Parsimony and the problem of inapplicables in sequence data," in *Parsimony, Phylogeny, and Genomics*, 2005, pp. 81–116.

[19] J. Hein, "A heuristic method to reconstruct the history of sequences subject to recombination," *J. Mol. Evol.*, vol. 36, no. 4, pp. 396–405, 1993.

[20] B. Chor, and T. Tuller, "Maximum likelihood of evolutionary trees is hard," in *Proc. Annu. Int. Conf. Res. Comput. Mol. Biol.*, 2005, pp. 296–310.

[21] N. Saitou, "Sequence homology handling," in *Introduction to Evolutionary Genomics*. London, U.K.: Springer, 2013, pp. 301–334.

[22] P. Lemey, M. Salemi, and A.-M. Vandamme, *The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[23] J. Felsenstein, "Evolutionary trees from gene frequencies and quantitative characters: Finding maximum likelihood estimates," *Evolution*, vol. 35, no. 6, pp. 1229–1242, Nov. 1981.

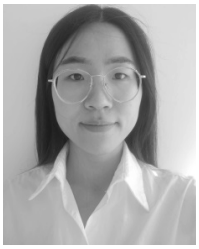
[24] J. H. Camin and R. R. Sokal, "A method for deducing branching sequences in phylogeny," *Evolution*, vol. 19, no. 3, pp. 311–326, Sep. 1965.

[25] J. Hein, "Reconstructing evolution of sequences subject to recombination using parsimony," *Math. Biosci.*, vol. 98, no. 2, pp. 185–200, Mar. 1990.

[26] S. Poe, "The effect of taxonomic sampling on accuracy of phylogeny estimation: Test case of a known phylogeny," *Hydrol. Process.*, vol. 28, no. 8, pp. 2945–2960, 1998.

[27] T. H. Jukes and C. R. Cantor, "Evolution of protein molecules," in *Mammalian Protein Metabolism*. 1969, ch. 24, pp. 21–132.

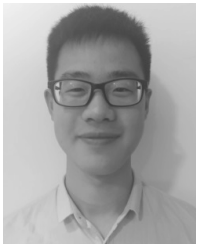
[28] M. Hasegawa, H. Kishino, and T.-A. Yano, "Dating of the human-ape splitting by a molecular clock of mitochondrial DNA," *J. Mol. Evol.*, vol. 22, no. 2, pp. 160–174, Oct. 1985.
[29] S. Tavaré, "Some probabilistic and statistical problems in the analysis of DNA sequences," *Lectures Math. Life Sci.*, vol. 17, no. 2, pp. 57–86, 1986.
[30] K. Tamura and M. Nei, "Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees," *Mol. Biol. Evol.*, vol. 10, no. 3, pp. 512–526, 1993.
[31] M. Kimura, "A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences," *J. Mol. Evol.*, vol. 16, no. 2, pp. 111–120, Jun. 1980.
[32] S. Santander-Jiménez and M. A. Vega-Rodríguez, "On the design of shared memory approaches to parallelize a multiobjective bee-inspired proposal for phylogenetic reconstruction," *Inf. Sci.*, vol. 324, pp. 163–185, Dec. 2015.
[33] J. G. Burleigh and S. Mathews, "Assessing systematic error in the inference of seed plant phylogeny," *Int. J. Plant Sci.*, vol. 168, no. 2, pp. 125–135, Feb. 2007.
[34] J. R. Macey, "Plethodontid salamander mitochondrial genomics: A parsimony evaluation of character conflict and implications for historical biogeography," *Cladistics*, vol. 21, no. 2, pp. 194–202, 2005.
[35] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator–based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, Aug. 2018.
[36] C. L. Hwang, and A. S. M. Masud, *Multiple Objective Decision Making—Methods and Applications*. 1994.
[37] Y. Wu, H. Huang, N. Wu, Y. Wang, M. Z. Alam Bhuiyan, and T. Wang, "An incentive-based protection and recovery strategy for secure big data in social networks," *Inf. Sci.*, vol. 508, pp. 79–91, Jan. 2020.
[38] W. Cancino, L. Jourdan, E.-G. Talbi, and A. C. B. Delbem, "A parallel multi-objective evolutionary algorithm for phylogenetic inference," in *Proc. Int. Conf. Learn. Intell. Optim.* Berlin, Germany: Springer, 2010.
[39] C. Zambrano-Vega, A. J. Nebro, and J. F. Aldana-Montes, "MO–phylogenetics: A phylogenetic inference software tool with multi-objective evolutionary metaheuristics," *Methods Ecol. Evol.*, vol. 7, no. 7, pp. 800–805, Jul. 2016.
[40] J. Jansson, C. Shen, and W.-K. Sung, "Improved algorithms for constructing consensus trees," *J. ACM*, vol. 63, no. 3, pp. 1–24, Jun. 2016.
[41] X. Min, M. Zhang, S. Yuan, S. Ge, X. Liu, X. Zeng, and N. Xia, "Using MOEA with redistribution and consensus branches to infer phylogenies," *Int. J. Mol. Sci.*, vol. 19, no. 1, p. 62, Dec. 2017.
[42] Y. Ju, S. Zhang, N. Ding, X. Zeng, and X. Zhang, "Complex network clustering by a multi-objective evolutionary algorithm based on decomposition and membrane structure," *Sci. Rep.*, vol. 6, Sep. 2016, Art. no. 33870.
[43] C. B. Pettey, M. R. Leuze, and J. J. Grefenstette, "Parallel genetic algorithm," in *Proc. 2nd Int. Conf. Genet. Algorithms*, 1987.
[44] A. R. Lemmon and M. C. Milinkovitch, "The metapopulation genetic algorithm: An efficient solution for the problem of large phylogeny estimation," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 16, pp. 10516–10521, Aug. 2002.
[45] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: New insights," *Struct. Multidiscpl. Optim.*, vol. 41, no. 6, pp. 853–862, Jun. 2010.
[46] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, "A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4831–4843, Jun. 2019.
[47] Y. Wu, H. Huang, Q. Wu, A. Liu, and T. Wang, "A risk defense method based on microscopic state prediction with partial information observations in social networks," *J. Parallel Distrib. Comput.*, vol. 131, pp. 189–199, Sep. 2019.
[48] D. Dong, W. Su, W. Shi, Q. Zou, and S. Peng, "VCSRA: A fast and accurate multiple sequence alignment algorithm with a high degree of parallelism," *J. Genet. Genomics*, vol. 45, no. 7, pp. 407–410, Jul. 2018.
[49] Q. Zou, Q. Hu, M. Guo, and G. Wang, "HAlign: Fast multiple similar DNA/RNA sequence alignment based on the centre star strategy," *Bioinformatics*, vol. 31, no. 15, pp. 2475–2481, Aug. 2015.
[50] L. Wei, P. Xing, J. Zeng, J. Chen, R. Su, and F. Guo, "Improved prediction of protein–protein interactions using novel negative samples, features, and an ensemble classifier," *Artif. Intell. Med.*, vol. 83, pp. 67–74, Nov. 2017.
[51] Q. Zou, S. Wan, X. Zeng, and Z. S. Ma, "Reconstructing evolutionary trees in parallel for massive sequences," *BMC Syst. Biol.*, vol. 11, no. 6, p. 100, 2017.
[52] L. Wei, S. Wan, J. Guo, and K. K. Wong, "A novel hierarchical selective ensemble classifier with bioinformatics application," *Artif. Intell. Med.*, vol. 83, pp. 82–90, Nov. 2017.
[53] S. Wan and Q. Zou, "HAlign-II: Efficient ultra-large multiple sequence alignment and phylogenetic tree reconstruction with distributed and parallel computing," (in English), *Algorithms Mol. Biol.*, vol. 12, p. 25, Sep. 2017.
[54] R. Guo, Y. Zhao, Q. Zou, X. Fang, and S. Peng, "Bioinformatics applications on apache spark," *GigaScience*, vol. 7, no. 8, p. giy098, 2018.
[55] L. Wei, P. Xing, R. Su, G. Shi, Z. S. Ma, and Q. Zou, "CPPred–RF: A sequence-based predictor for identifying cell–penetrating peptides and their uptake efficiency," *J. Proteome Res.*, vol. 16, no. 5, pp. 2044–2053, May 2017.
[56] M. Zaharia, C. Mosharaf, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," *HotCloud*, vol. 10, no. 10, p. 95, 2010.
[57] C. Li, T. Wen, H. Dong, Q. Wu, and Z. Zhang, "Implementation of parallel multi-objective artificial bee colony algorithm based on spark platform," in *Proc. 11th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Aug. 2016, pp. 592–597.
[58] P. O. Lewis, "A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data," *Mol. Biol. Evol.*, vol. 15, no. 3, pp. 277–283, Mar. 1998.
[59] C. Zambrano-Vega, A. J. Nebro, and J. F. Aldana-Montes, "MO–Phylogenetics: A phylogenetic inference software tool with multi-objective evolutionary metaheuristics," *Methods Ecol. Evol.*, vol. 7, no. 7, pp. 800–805, Jul. 2016.
[60] T. Song, A. Rodriguez-Paton, P. Zheng, and X. Zeng, "Spiking neural P systems with colored spikes," *IEEE Trans. Cogn. Develop. Syst.*, vol. 10, no. 4, pp. 1106–1115, Dec. 2018.
[61] T. Song, X. Zeng, P. Zheng, M. Jiang, and A. Rodriguez-Paton, "A parallel workflow pattern modeling using spiking neural p systems with colored spikes," *IEEE Trans. Nanobiosci.*, vol. 17, no. 4, pp. 474–484, Oct. 2018.
[62] T. Song, L. Pan, T. Wu, P. Zheng, M. L. D. Wong, and A. Rodriguez-Paton, "Spiking neural P systems with learning functions," *IEEE Trans. Nanobiosci.*, vol. 18, no. 2, pp. 176–190, Apr. 2019.
[63] T. Song, S. Pang, S. Hao, A. Rodríguez-Patón, and P. Zheng, "A parallel image skeletonizing method using spiking neural P systems with weights," *Neural Process Lett.*, vol. 50, no. 2, pp. 1485–1502, Oct. 2019.
[64] X. Zhang, Q. Zou, A. Rodriguez-Paton, and X. Zeng, "Meta–path methods for prioritizing candidate disease miRNAs," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 1, pp. 283–291, Jan. 2019.
[65] Q. Zou, J. Li, L. Song, X. Zeng, and G. Wang, "Similarity computation strategies in the microRNA-disease network: A survey," *Briefings Funct. Genomics*, Jul. 2015, Art. no. elv024.
[66] Z. Hong, X. Zeng, L. Wei, and X. Liu, "Identifying enhancer-promoter interactions with neural network based on pre-trained DNA vectors and attention mechanism," *Bioinformatics*, Sep. 2019.
[67] F. G. C. Cabarle, R. T. A. De La Cruz, D. P. P. Cailipan, D. Zhang, X. Liu, and X. Zeng, "On solutions and representations of spiking neural P systems with rules on synapses," *Inf. Sci.*, vol. 501, pp. 30–49, Oct. 2019.
[68] X. Zeng, Y. Zhong, W. Lin, and Q. Zou, "Predicting disease-associated circular RNAs using deep forests combined with positive-unlabeled learning methods," *Briefings Bioinf.*, Oct. 2019.
[69] X. Zeng, S. Zhu, X. Liu, Y. Zhou, R. Nussinov, and F. Cheng, "deepDR: A network-based deep learning approach to in silico drug repositioning," *Bioinformatics*, May 2019.



QIANQIAN ZHANG received the B.E. degrees from the College of Science, Anhui University of Science and Technology, in 2017. She is currently pursuing the master's degree with the School of Informatics, Xiamen University. Her research interests are phylogeny and data analysis.

JUN ZHANG is currently the Vice President of the Heilongjiang Provincial Agricultural Reclamation General Hospital, where he is engaged in rehabilitation therapy research. He is also a Graduate Tutor with Harbin Medical University and serves as the Deputy Director of the Trauma Professional Committee of China Rehabilitation Medicine Association. His research interest is rehabilitation.

YUE ZHONG received the B.E. degree in network engineering from Fujian Normal University, Fuzhou, China, in 2018. She is currently pursuing the master's degree in computer science with Xiamen University. Her research interests are bioinformatics and machine learning.



CONGMING YE received the B.E. degrees from the College of Science, Fujian Normal University of Software Engineering, in 2018. He is currently pursuing the master's degree with the School of Informatics, Xiamen University. His research interests are bioinformatics and natural language processing.

XIAOPING MIN is currently an Associate Professor with the School of Informatics, Xiamen University. His research interests are distributed databases and software engineering.