A Parallel Shuffled Frog Leaping Algorithm Based on Stem Regions Combinatorial Optimization for RNA Secondary Structure Prediction

Chunlei Liu
Department of Computer Science
and Engineering
Harbin Institute of Technology
150001 Harbin, China
812liuchunlei@163.com

Zhenzhou Ji
Department of Computer Science
and Engineering
Harbin Institute of Technology
150001 Harbin, China
jzz@pact518.hit.edu.cn

Yingsen Hong
Department of Computer Science
and Engineering
Harbin Institute of Technology
150001 Harbin, China
hongyingsen@139.com

***Abstract*—RNA Secondary Structure Prediction is an important part of the biological computing. RNA secondary structure prediction algorithms tend to have higher time and space complexity. Some swarm intelligence algorithms can also be applied to RNA secondary structure prediction on the basis of stem regions combinatorial optimization algorithm, such as genetic algorithm (GA), particle swarm optimization algorithm (PSO) and shuffled frog leaping algorithm (SFLA). And these algorithms achieved good effects. According to shuffled frog leaping algorithm in the application of RNA secondary structure prediction, this paper presents a parallel discrete shuffled frog leaping algorithm (parallel-DSFLA). This parallel algorithm can run on a distributed cluster system using the MPI programming mode. The experimental results show that the parallel-DSFLA got better speed-up ratio, can improve the RNA secondary structure prediction efficiency and save time.**

Keywords-RNA secondary structure prediction; MPI; parallel discrete shuffled frog leaping algorithm; swarm intelligence

I. INTRODUCTION

RNA (Ribonucleic Acid) is one of the most important molecules in the biological system. RNA primary structure is a polynucleotide chain connected by four kinds of bases through 3', 5'phosphodiester bond. RNA is generally single-stranded linear molecule, it tends itself folded back to form a double-stranded secondary structure, and further folded to form a tertiary structure. RNA is the carrier of transcription and translation between DNA and protein, the research of RNA function is currently an important topic in biology, and structure is an important factor to function. Determination of RNA tertiary structure by experiment is complicated and costs too much time. Meanwhile, RNA tertiary structure is difficult to predict directly by primary structure, so secondary structure prediction is the only way to predict tertiary structure [1] and it is particularly important to predict RNA secondary structure through calculation methods. At present all kinds of RNA database have accumulated a large amount of RNA sequence data. With the RNA sequence data as input, computer can predict the corresponding secondary structure based on various calculation models, and then analyses RNA function.

RNA secondary structure prediction methods are divided into two categories: method based on comparative sequence analysis and method based on the minimum free energy [2]. When there is no prior knowledge, only given RNA primary structure, RNA secondary structure prediction generally uses the minimum free energy model. Among these algorithms, Nussinov algorithm and Zuker algorithm [3] is dynamic programming algorithm. Without considering pseudoknots, their time complexity is O (n^3). The algorithm proposed by Rivas and Eddy can predict pseudoknot and it requires O (n^6) time complexity [4]. Stem regions combinatorial optimization algorithm is another method based on the minimum free energy. Although it was proved to be NP - difficult problem [5], it can get the optimal solution more quickly with the development of swarm intelligence optimization algorithm. Such as genetic algorithm, ant colony algorithm and particle swarm optimization algorithm have been applied to RNA secondary structure prediction.

RNA secondary structure prediction algorithm generally has high complexity and large amounts of calculation. When RNA sequence is longer, to get the optimal structure takes more time. Therefore, to improve the performance of the algorithm through effective parallel methods has great significance. On the basis of literature [6], combining with the parallel computing theory, we designed a MPI-based parallel shuffled frog leaping algorithm, which can effectively improve the efficiency of RNA secondary structure prediction.

II. BASIC KNOWLEDGE

RNA consists of four kinds of bases, namely A (adenine), C (cytosine), G (guanine), U (uracil). A single-stranded RNA can be viewed as a n-length sequence from the alphabet {A, C, G, U}, $R=r_1r_2\cdots r_n$, $r_i\in$ {A, C, G, U}. RNA tends itself folded back to form different substructures following the principle of complementary base pairs (A-U, C-G, U-G), such as stack, hairpin loop, bugle, internal loop and multiple loop. A-U pairing, C-G pairing and G-U pairing are collectively referred to as standard base pairs.

Stem Region: $R1=r_ir_{i+1}\cdots r_{i+k-1}$ and $R2=r_{j-k+1}r_{j-k+2}\cdots r_j$ are two subsequences of R and (r_{i+t}, r_{j-t}) is a standard base pair, t=0,1,…,k-1, then R1 and R2 constitute a stem region in R's secondary structure, recorded as S (i, j, k), k is the

Published by Atlantis Press, Paris, France.

length of stem region. And different stem region have different free energy.

If two stem regions S1 (i_1, j_1, k_1) and S2 (i_2, j_2, k_2) neither happen to overlap, also do not cross, we say the two stem regions are compatible.

S = {S1, S2, … , Sm} is a set of R's stem regions and R is a n-length RNA sequence. If any two of the stem regions are compatible, S set can uniquely determine R's secondary structure. And the number of stem in S is less than (n-2)/7.

According to above definitions and properties, we can transform RNA secondary structure prediction to a compatible stem regions combination optimization problem. That is, choose a combination from a stem region set which RNA sequence constitute, and make sure that it has the minimum free energy.

III. SFLA and its Application in RNA Secondary Structure Prediction

A. SFLA

Shuffled Frog Leaping Algorithm (SFLA) is proposed to solve the combinatorial optimization problem of water supply network by Eusuff and Lansey in 2003 [7]. It is a new swarm intelligence algorithm which combines memetic algorithm based on genetic behavior with PSO algorithm based on social behavior and simulation of foraging behavior [8]. The algorithm is simple in concept, has less adjustment parameters and higher calculation speed. It has better global search optimization ability and is easy to implement.

Thought of SFLA: A group of frogs are randomly selected from wetland to search for food. All the frogs are divided into several subgroups. Each subgroup has its own culture and performs each local search strategy. Each individual in one subgroup affects others, but also is affected by other individuals. And they evolve with the evolution of the subgroup. When all subgroups evolve to a certain extent, they exchange the information with others. Finally the frogs evolve towards the best direction.

When a frog's fitness value increases, it immediately returns to the subgroup, making the improvement of information can immediately provide service for the subsequent evolution. This is the advantage of SFLA compared with GA. SFLA is similar to PSO in local search process, but it puts more emphasis on local search capabilities. As a new swarm intelligence optimization algorithm, SFLA shows stronger local search ability and better global search capability.

The basic process of the algorithm is as follows:

1) Initialize the population and make every frog has a vector X=(x_1, x_2, … , x_V) and a fitness value. Set the number of subgroup and the number of frogs in each subgroup.
2) Randomly generate initial frog groups, calculate the fitness value of each frog.
3) According to the fitness value, sort the frogs in descending order. Get the best global solution Px, and the frogs are averagely divided into several subgroups.
4) Search in each subgroup, to find the best and worst fitness value of the frog, which are defined as Pb and Pw. And then update the position of the worst frog.
5) Mix all subgroups
6) Judge stop condition, which can be a maximum number of iterations, or meet the convergence conditions. If meet the condition, output optimal solution, otherwise return to step 3).

Update strategy in each subgroup:

$$\mathrm{Di} = \mathrm{rand}(\,) * (\mathrm{Pb} - \mathrm{Pw}) \qquad ①$$

New position:

$$\mathrm{Pw} = \mathrm{Pw} + \mathrm{Di}, \mathrm{Dmax} \geq \mathrm{Di} \geq -\mathrm{Dmax} \qquad ②$$

rand ( ) is a random number between 0 and 1. Dmax is the maximum range frog moves. If the above operation will produce a better solution, replace the original fitness value of the worst frog with the better value. Otherwise, replace the Pb with Px in formula ①, and then calculate a new solution according to the formula ① and ②. If fitness value is still not improved, randomly generate a new solution to replace the worst frog.

B. Application in RNA Secondary Structure Prediction

RNA secondary structure prediction is to seek a combination of stem regions which has the minimum free energy. The solution space of the problem is discrete. To use SFLA, we need to define special, discrete representation and operators. Each dimension of each frog's vector represents a stem region, so each frog represents a set of a group of the stem regions. The movement of frog redefined as addition and deletion of stem regions in the set. To add stems exist in Pb, so as to move closer to the optimal solution. To prevent algorithm from converging too fast to get the optimal solution, first delete the original stems which do not exist in Pb before adding. Thus formula ① and ② are redefined as follows:

$$\mathrm{Pw} = \mathrm{Pw} - \mathrm{O} \qquad ③$$

$$\mathrm{Pw} = \mathrm{Pw} + \mathrm{C} \qquad ④$$

In Formula ③, O is the set of stems which are chosen to delete. In Formula ④, C is the set of stems which are chosen from Pb to add. This is the application of discrete shuffled frog leaping algorithm in RNA secondary structure prediction.

IV. Parallel Analysis and Design

A. Analysis of Serial Program Hotspots

Literature [6] has proved that DSFLA in RNA secondary structure prediction has a good effect, but there are still some places can be improved. The larger the number of frog is, the greater the possibility to get global optimal solution is. But the calculation quantity also increases and it will take more time. Larger number of frogs in subgroup will also lead to the extension of the time to search. When RNA sequence is longer and the number of stems in stem region pool is more, the dimension of solution is more. It will take more time to get the optimal solution. All above these are factors that reduce prediction efficiency.

We found that during the evolution of each subgroup, they have no influence on others and all of them search independently. When each subgroup's evolution finishes, mix all subgroup and exchange information with each other. This is the basis of the parallel algorithm in this paper. Different processes perform the search of different subgroups in parallel, so as to achieve the purpose of shortening the global search time and improving the search efficiency. Firstly we analyze serial SFLA to find the algorithm hotspots. In order to improve overall performance effectively, we can start parallel design from the hotspots.

Performance testing tool: Intel(R) VTune(TM) Performance Analyzer

Hardware environment: Intel (R) Core (TM) 2 Quad CPU, main frequency 2.50 GHz; 2 GB memory.

Operating system: centos 6.3.

The results of serial algorithm performance analysis are shown in TABLE I.

TABLE I. SERIAL ALGORITHM PERFORMANCE ANALYSIS TABLE

Function	Calls	Time(V=50)	Time(V=100)	Time(V=200)
fitness()	200	84	141	227
init()	1	1368	1804	3815
sort()	200	268218	435405	575549
sortPop()	80000	105752	153171	193363
update()	80000	819836	1373332	2234739
copy()	200	14208	29016	46679
report()	200	5126	4878	3823
main()	1	1110047	1870490	2866986

Total number of iterations: G=200, number of frog: P=200, number of subgroup: M=20, number of frogs in Subgroup: I=10, dimension: V=50, 100, 200, number of iterations in the subgroup: N=20

Observing from the results of analysis, we found that sortPop ( ) and update ( ) have the largest number of calls in the algorithm. Function sortPop ( ) is to sort the frogs in subgroup, and then get Pb and Pw. Function update ( ) is to update the information in subgroup. They are used to process the data of subgroup. Function sort ( ) is to sort all the frogs to find out the global best solution Px, it need to process a large quantity of data, so it takes much time, too. Function update ( ) occupied the most time of the whole procedure, it is the hotspot of this algorithm. Next we will design parallel algorithm to reduce the running time of the program.

B. Parallel Algorithm Design

In this paper, the parallel method is based on message passing interface (MPI). The parallel algorithm can run on a distributed cluster system. Different processes of different nodes perform independent tasks in parallel. Each parallel process has independent storage space, they exchange date with each other by sending and receiving messages explicitly [9].

When applying DSFLA to predict RNA secondary structure, every subgroup has the same number of frogs and iterations, they perform the same operation, so each process has the same operating load. We chose static load plan in this MPI parallel algorithm design, it can avoided some additional communication overhead which dynamic load program costs.

Communication overhead of parallel algorithm exists between the master process and sub processes. The master process is responsible for initializing population and processing global data. When all the data of population have been processed, the master process divides them into P copies averagely according to certain rules and sends to P sub processes. After receiving the data, each sub process begins its task with no influence from each other. So there is no communication overhead between each sub process. When each sub process finishes its task, they send their data back to the master process. The master process receives and mixes all the global data. And then the master process determines whether to perform the next iteration.

Parallel program flow chart is shown in Figure 1.

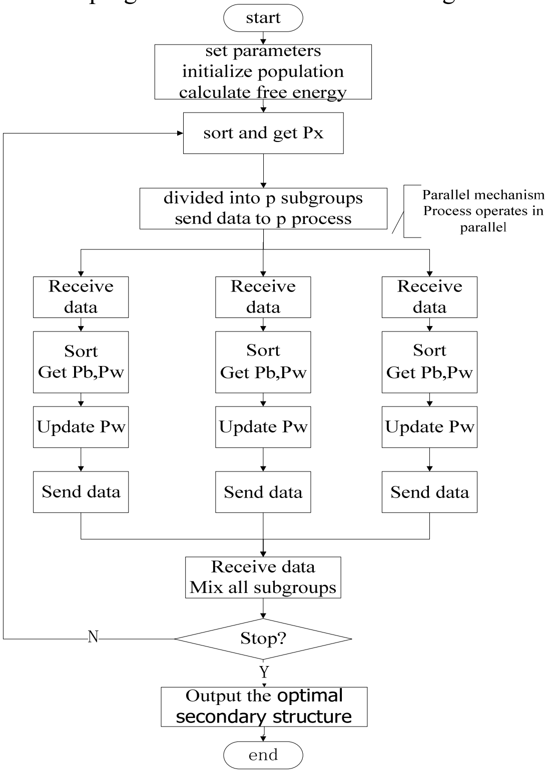

Figure 1. Flow chart of parallel algorithm

The specific design idea is as follows:

The algorithm of master process:

1) Set parameters. Randomly generate sets of stem regions which are compatible between each other. These sets constitute the initial population.
2) Calculate the free energy of every frog and sort them in descending order. Get the frog that has the minimum free energy, recorded as Px.

3) Divide population into P copies averagely and send to P sub processes.
4) Receive data from P sub processes and mix all the data to form a new population.
5) Judge stop condition. If meet the condition, stop and output the combination of stem regions that has the minimum free energy, otherwise return to step 2).

The algorithm of sub process:

1) Receive data from the master process.
2) According to free energy, sort the frogs in descending order. Find the maximum and minimum free energy of the frog, defined as Pb and Pw.
3) According to the formula ③ and ④, update Pw.
4) If reaching the number of iterations, send data back to the master process. Otherwise return to step 2).

V. Effect of Parallelization

Speed-up ratio Sn represents the improved degree of the running time in parallel algorithm. It is an important standard to evaluate performance of parallel algorithm [10]. The running time of the serial algorithm is T_S and the running time of the parallel algorithm is T_P. Then speed-up ratio Sn is as follow:

$$\text{Sn} = T_S \,/\, T_P$$

Comparing the running time of serial algorithm with parallel algorithm is in the case of the same parameters.

TABLE II shows the speed-up ratio with four processes running in parallel. One of the processes is master process, it performs the global operation. Subgroups are averagely divided into three parts, and they are processed by the other three processes.

TABLE III shows the speed-up ratio with eight processes running in parallel.

The parameters are as follow: total number of iterations: G=200, number of frog: P=200, number of subgroup: M=20, number of frogs in subgroup: I=10, dimension: V=50, 100, 200, number of iterations in the subgroup: N=20.

TABLE II. Analysis of Speed-up Ratio (Four Processes)

Dimension	DSFLA Runtime(s)	Parallel-DSFLA Runtime(s)	Speed-up ratio
50	0.9	0.45	2
100	1.38	0.72	1.92
200	2.23	1.12	1.99

TABLE III. Analysis of Speed-up Ratio (Eight Processes)

Dimension	DSFLA Runtime(s)	Parallel-DSFLA Runtime(s)	Speed-up ratio
50	0.9	0.33	2.7
100	1.38	0.51	2.71
200	2.23	0.83	2.67

The parallel algorithm presented in this paper is a partial parallel parallelization. It only parallels the operation of subgroup, which consumes the most time. Other global operation is performed serially by the master process. Because of the communication overhead, the speed-up ratio cannot reach the theoretical highest value n (n is the number of processors). Even so, the parallel algorithm achieved good speed-up ratio, and the speed-up ratio increases as the number of parallel processes increases. When RNA sequence is longer, the number of stem region is more and dimension of solution space is higher, parallel algorithm can save more time than serial algorithm.

VI. Conclusion

In this paper, we studied the application of discrete shuffled frog leaping algorithm in RNA secondary structure prediction. On the basis of it, we study and design parallel discrete shuffled frog leaping algorithm using MPI parallel programming model. The experiment results show that this algorithm can solve the high-dimensional combination optimization problem more effectively. When parallel-DSFLA is applied to predict long RNA secondary structure, it not only ensures better performance of DSFLA in prediction, but also improves the prediction efficiency and saves more time. Above all, the parallel algorithm for RNA secondary structure prediction is of great significance.

Acknowledgment

This research was supported by the National Natural Science Foundation of China under Grant No. 61173024.

References

[1] Tao Jiang, Ying Xu, Michael Q. Zhang. Current Topics in Computational Molecular Biology [M]. Beijing: Tsinghua university press, 2002.

[2] Gardner PP, Giegerich R. A comprehensive comparison of comparative RNA structure prediction approaches [J]. BMC Bioinformatics, 2004. 1−32.

[3] Zuker M. On finding all suboptimal foldings of an RNA molecular [J]. Science, 1989, 244(4900): 48-52.

[4] E Rivas, S Eddy. A Dynamic Programming Algorithm for RNA Structure Prediction Including Pseudoknots [J]. Journal of Molecular Biology, 1999, 285: 2053- 2065

[5] Lyngsa R B, Pedersen C N. Pseudoknots in RNA secondary structures [C]. New York: ACM, 2000: 201-209.

[6] Juan Lin, Yiwen Zhong, Jun Zhang. Discrete Shuffled Frog Leaping Algorithm for RNA Secondary Structure Prediction [J]. Journal of Nanjing Normal University, 2011, 11(4): 63-69.

[7] Muzaffar M. Eusuff, Kevin E. Lansey. Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm [J]. Journal of Water Resources Planning and Management, 2003, 129 (3): 210-225.

[8] Yi Han, Jianhu Cai, Gengui Zhou, et al. Advances in Shuffled Frog Leaping Algorithm [J]. Computer Science, 2010, 37 (7): 16-19.

[9] Kai Huang, Zhiwei Xu, Scalable Parallel Computing Technology, Architecture, Programming [M].Beijing: China Machine Press, 2000.

[10] Guolinag Chen, Design and Analysis of Parallel Algorithm [M], Beijing: Higher Education Press, 2009.

Published by Atlantis Press, Paris, France.