I.J. Engineering and Manufacturing **2011, 2, 48-53**
Published Online April 2011 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijem.2011.02.08

Available online at http://www.mecs-press.net/ijem

An Efficient Genetic Algorithm Orienting to the Protein Fold Prediction

Xiangting Fan [a], Zhenzhou Ji [a]

[a] *Department of Computer Science and Engineering, Harbin Institute of Technology, 150001 Harbin, China*

Abstract

Proteins are amino acid chains that acquire their biological and biochemical properties by folding into unique 3-dimensional structures. The biological function of a protein is dependent on the protein folding into the correct, or "native", state. At present, there are so many ideas to predict the structure of the protein folding. This paper first present the concept of protein folding and how is significant to study protein fold prediction. In this paper we join the simulated annealing factor into Parallel Genetic Algorithm and use this hybrid Parallel GA to predict the structure of protein fold. The revised algorithm is more efficient than traditional Genetic Algorithm and simulated annealing algorithm.

Index Terms: biological function; structure prediction of protein folding; parallel genetic algorithm; sumulated annealing factor; revised

1. Introduction (Heading 1)

Proteins are the biological molecules that are the building blocks of cells and organs, and the biochemical processes required to keep living organisms alive are catalyzed and regulated by a particular category of proteins called enzymes. Protein engineering is the introduction of modern bio-technology field, and its fundamental purpose is to transform the naturally occurring proteins according to people's ideas, or to design a non-natural new protein with certain special functions according to need, and one important basis of this transformation and design is the prediction of protein folding structure. The shapes of protein fold structure largely determine the biological function that they may have, which means that there is consistent between structures and functions of proteins. Therefore, the research and prediction of protein folding structure has an extremely important position in protein biological engineering.

As understanding of the details of protein structure and their folding rule becomes more and more deeper, there are two difficult problems about protein folding prediction problem in general. First, how to attribute the

* Corresponding author.
E-mail address:

mathematical models that can better reflect the interaction between amino acid residues and environmental conditions and so on; And second, how to develop more efficient search method for the functional structure of the search.

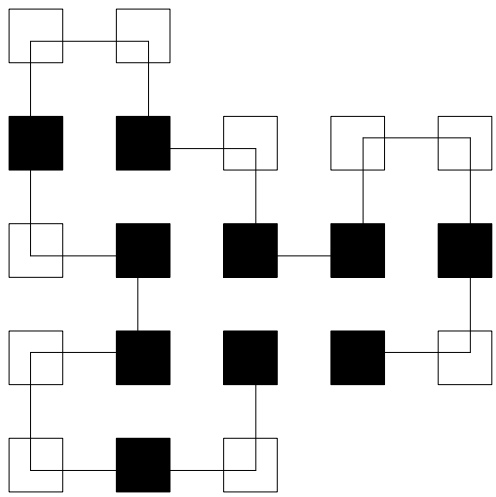

Fig. 1. Minimum energy conformations dimensional trellis of the squence length 20

K.A. Dill established Hydrophobic-Hydrophilic model according to the protein structure features that the hydrophobic residues bury under within the protein molecule, the hydrophilic residues exposed to the protein molecules in contact with water features on the surface, the HP model, as described in [1]. In the HP model, H stands for hydrophobic amino acids, and P stands for hydrophilic amino acids. So that the protein chain can be represented by a finite length of the string on the alphabet {H, P}, such as the protein chain PHPHHPHPHPHH. Each amino acid can be imaged as a node. The values of the energy between HH, HP, PP residue fulfill some certain conditions. In the two-dimensional space, we should take the value of the energy as following: E_{HH} =- 1, E_{HP} = E_{PP} = 0. Fig. 1 (Small black box: H; Small white box: P)is the minimum energy conformations dimensional trellis of the sequence length 20 (HPHPPHHPHPPHPHHPPHPH) . Energy function is -9.

2. Simulated annealing factor adding parallel genetic algorithm

Genetic algorithms (GA) is an effective search methods of the same category based on natural selection and the principle of genetics, which starts from a population using selection, crossover and mutation operators to evolve the populations continual, and finally obtain the global optimal solution, more details are shown in [2]. Parallel genetic algorithm adds the spatial structure factors of the natural biological populations in the implementation of the genetic algorithm, the aim is that the algorithm should run on massively parallel machines. Even though the parallel genetic algorithm is implemented by sequence types, genetic algorithm can overcome the classic lack of the performance in classic genetic algorithm , thereby enhancing the efficiency of the algorithm.

We also found that in the practical application traditional genetic algorithm can produce the phenomenon of premature convergence easily, the objective function values of a small number of individuals during the group far weigh the objective function values of other individuals, so the probability that they participate in the selection copy operation is larger, the effect by the cross, mutation little. Then, after a few iterations, the individual will fill up the entire group, and that cause evolution process early convergence. Aiming at these

problems we should add simulated annealing factor in parallel genetic algorithm, as described in [3][4]. Algorithm described as:

begin
Initialization
A randomly generated initial population
Divide all the individuals into p subpopulation; Set the number of independent evolution as S, set the top evolution algebra as LOOP1;
Set the initial temperature temp0 of the simulated annealing algorithm and the convergence rate a: temperature $temp_{i+1} = a * temp_i$, $0 < a < 1$, *gradually reduced, subscript i represents the generation of the individual;*
Set the number of variables plot LOOP2 = 0
for i=1 to p par-do
Evaluation of the child population in the objective function of each individual $cost(X_j)$;
while loop2 < S do
for j=1 to n do
selection, crossover, mutation
calculate the fitness
end for;
LOOP1= LOOP1+1,LOOP2 =LOOP2+1;
end while
end for
Select some individuals as the migrants
Send emigrants and receive immigrants
If LOOP1<MAXLOOP, then $T_i = a*T_i$, *go to (2)*
end;

3. Prediction of protein folding

For specific prediction of protein folding, the algorithm implementation steps are as follows, more detailed shown in[5]:

3.1. Coding

Enter the HP chain of the protein sequence that need to be folded, and the total number of the chain is lchrom. Generate random numbers 0,1, the total number is 2*(lchrom-1); The direction of movement is decided by amino acid sequence 0,1. Described as follows, 00: right; 11: left; 10: down; 01: up.

3.2. Determine the initial populations

For proteins to be processed, because its primary structure is known, the hydrophilic or hydrophobic properties of all the elements of its amino acid sequence are definite. A lot of experiments show that the population is more bigger, and the individual species the population contains are more abundant, the program will finally get the better solution. To ensure the diversity of initial population solutions, we have adopted a completely random generation method for information for characterizing the moving direction of the next element. Note here a little that due to the amino acid sequence started in the flat space, the individual sequences may lead to some illegal in the plane overlap occurred in the production. In initialized, the program should estimate whether the individuals are legitimate, if not legally then require additional penalty factor.

3.3. Calculation of fitness

As when calculate the fitness values, the order and locations of each amino acid must be obtained, the program sets up two (2*lchrom-1) * (2*lchrom-1) matrixes, Sample[][] storing 1,, 2,3 , ... folding sequence and SetArray[][] storing H or P. When SetArray[][] matrix appears adjacent HH but in the Sample [][] matrix they are not linked together, then we write the energy to be -1, more information described in [6]. Count all this phenomenon, we can get the value of E. Judge the total number of the figures appearing in the matrix, denoted by d. .If d!=lchrom, then add a penalty factor that decreases the probability of the E is selected. The penalty factor is added as in:

$$E = \sum_{i<j} E\sigma_i\sigma_j\Delta(r_i - r_j) \times P. \quad (1)$$

In (1),when d=lchrom, P=1;when d != lchrom, then p take a negative. As the value of the energy is negative, then when the energy value is multiplied by a negative number, the energy becomes positive. Absolutely, the value of the energy is increased. So far, the task of making a model is completed.

3.4. Operation of genetic operators

The selection operators are operated in groups according to the roulette selection method. According to the crossover probability cross-operation. Mutation probability according to a randomly selected site on the role of string mutation operator. In the crossover operator and mutation operator ,algorithm no longer just simple preserved better individuals, but to add simulated annealing factor, retention of some poor individual, so that the conformation of populations diversity and fitness has increased, to avoid falling into local optimum.

3.5. Parallel Implementation Algorithms

This paper is analyzing and processing from the coarse-grained parallelism of genetic algorithms, as described in [7]. The population is divided into multiple subpopulations, each subpopulation evolves isolated and exchanges individuals occasionally. “Fig 2” can be used to represent the implementation process. Select some individual as the migrants to pass selection, crossover and mutation operation. Then send emigrants and receive immigrants. In addition, the two processes that send and receive immigrants are capable of running to avoid deadlock in data communication.

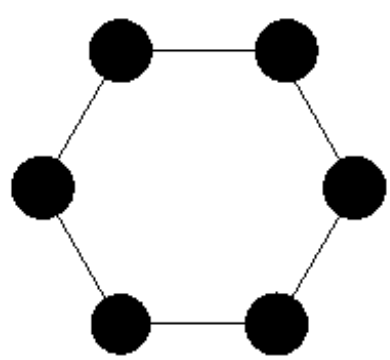

Fig. 2 Coarse grained parallel GA

There are an IBM 24-core server 3850 in our laboratory. Then the algorithm can come true parallel implementation in this server. And MPI communication is used to achieve parallelism, more details shown in [8]. SPMD model is needed. SPMD is described as Fig. 3.

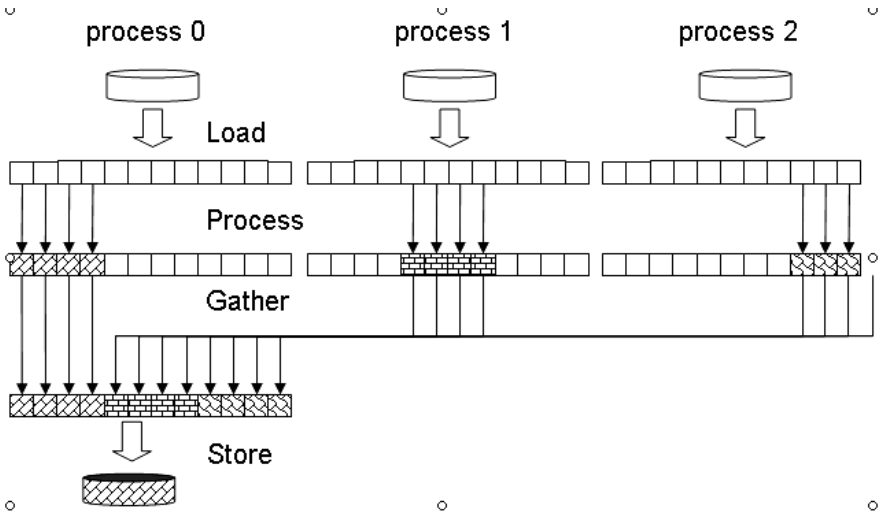

Fig. 3 The process of SPMD parallel program execution

4. Analysis

One of the most promising choices to make GA faster is to use parallel implementations. There are two reasons for parallelization, one is the nature, another is GA itself.

Take a number of protein sequence of different length as the test set, first of all sequences are done for HP by amino acid residues of the hydrophilic and hydrophobic, and then use the genetic algorithm adding annealing simulation factor to simulate annealing simulation factor, and in the process of simulation, parallel implementations are integrated into.

5. Conclution

In this paper, aiming at the problem of prediction of protein folding, a hybrid parallel genetic algorithm method is put forward, and simulated annealing factor is added to avoid the algorithm falling into local optimal solution. Although protein folding problem is a NP problem, with the deepening of understanding about mechanisms of protein folding, the continuous improvement of the algorithm and the continuous development of the computer, protein fold prediction problem would be better resolved, something new in [9].

References

[1] H.S. Chan ,K.A. Dill, Energy landscapes and the collapse dynamics of homopolymers, J. Chen. Phys, 1993, 99(3) , pp. 2116–2427.

[2] J. H. Holland, Adaptation in Nature and Artificial Systems. The University of Michigan Press, 1975.

[3] Lishan Tang, Non-numerical computation-Simulated Annealing Algorithm, 1rd ed. Being Jing: The Science Publishing Company, 1998. (in Chinese)

[4] H. P. Hsu, V. Mehra. W.Nadler, P,Grassberger, "Growth algorithms for lattice heteropolymers at low temperatures," Chem. Phy,2003(1).
[5] J. B. Martin, J. Gareth and W. Peter, "GENFOLD: A parallel genetic algorithm for protein folding prediction using the 3D-HP Side Chain model," Evolutionary Computation,2009,CEC'09, pp,1297–1304.
[6] H. Li,R. Helling, C. Tang, "Emergence of preferred structures in a simple model protein folding," Science, 1996, 273, pp. 666–669.
[7] W. J. Jiang, D.T. Luo, Y.S. Xu, X.M.Sun, Hybrid genetic algorithm research and its application in problem optimization, Intelligent Control and Automation,2004, pp. 2122–2126, Vol.3.
[8] Shamenm. Akhter, Jason. Roberts, Multi-Core Programming: Increasing performance through software multi-threading, publishing house of electronics industry, 2007.
[9] Dehzang, A. Khosravi and B. G. Fac. Introducing novel physicochenmical based features to enhance protein fold prediction accuracy, ICCDA, 2010, pp. V1-592–V2-596.
[10] V. Sharma, V. R. I. Kaila, and A. Annila, "Protein folding as an evolutionary process", Physica A, 2009, 388(6), pp. 851–862.