Improved Particle Swarm Optimization Algorithm for 2D Protein Folding Prediction

Xiaolong Zhang
Wuhan University of Science and Technology
School of Computer Science and Technology
Wuhan, China
xiaolong.zhang@mail.wust.edu.cn

Tingting Li
Wuhan University of Science and Technology
School of Computer Science and Technology
Wuhan, China
tingting.li@mail.wust.edu.cn

***Abstract*—One of the main problems of protein folding prediction is optimization computation. On basis of toy model, this paper proposes an improved particle swarm optimization algorithm for protein folding prediction. The algorithm introduces a new architecture that is characterized by balancing exploration and exploitation capability of particle swarm optimization algorithm. In the architecture, the population in each generation consists of three parts: an elitist part, an exploitative part, and an explorative part. In the meantime, it makes protein folding prediction more effective with the global search and local search ability in the improved algorithm. Furthermore, we have applied the improved particle swarm optimization algorithm to sequences with up to 55 monomers within toy model and the experimental results are compared with the global energy minimum reported in some literatures. It demonstrates that the proposed algorithm is effective to search for the native state of proteins with the lowest free energy.**

Keywords-protein folding, toy model, exploration, exploitation particle swarm optimization

I. INTRODUCTION

Protein folding prediction is a central problem in molecular and computational biology with the potential to reveal the function and behavior of proteins, which would greatly influence many areas in biology and medicine. Therefore, the prediction and research of protein folding structure plays a key role in protein engineering. In protein folding prediction, the difficulty of the folding problem lies in two major aspects. One is how to choose the potential energy function that can correctly distinguish the native states from non-native states of protein molecules. The other is how to select an effective method to search the global optimal solution by the given energy function defined by the first aspect.

In recent years, there are varieties of models proposed for protein folding problem, such as HP lattice model [1] and toy model [2]. In the HP lattice model, each amino acid residue is classified as an H (hydrophobic or non-polar) or a P (hydrophilic or polar). The residues must be placed on a two-dimensional grid without overlapping, so that adjacent amino acids in the sequence remain horizontally or vertically adjacent in the grid. The model considers only the interactions between neighboring non-bonded H monomers rather than the other non-local effects caused by P-P, H-P, and non-neighbored H-H pairs. Compared with HP lattice model, toy model considers the interaction among all resides and captures the main features and attributes of the protein folding state, then, it reflects more realistically the native states of proteins. In toy model, amino acid residues are grouped into two kinds: A (hydrophobic or non-polar) and B (hydrophilic or polar). There is only one bond between two consecutive residues, but the angle between the two bonds can change freely. For any n-mer chain the intra-molecular potential-energy function Φ can be described as :

$$\Phi = \sum_{i=2}^{n-1} V_1(\theta_i) + \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} V_2(r_{ij}, \xi_i, \xi_j) \tag{1}$$

V_1 is the bending potential and independent of the protein sequence as following:

$$V_1(\theta_i) = \frac{1}{4}(1 - \cos\theta_i) \tag{2}$$

And V_2 is the Lennard-Jones potential with a species-dependent coefficient $C(\xi_i, \xi_j)$ as following:

$$V_2(r_{ij}, \xi_i, \xi_j) = 4(r_{ij}^{-12} - C(\xi_i, \xi_j) \cdot r_{ij}^{-6}) \tag{3}$$

Where $C(\xi_i, \xi_j)$ for AA, BB and AB pairs is +1, $+1/2$, and $-1/2$ respectively, producing an intra-molecular mix of strong attraction, weak attraction and weak repulsion. r_{ij} denotes the distance between molecular i and j of the chain.

Even though toy model are highly simplified, it remains NP-hard [3] to solve the corresponding protein folding problem so that it is far from trivial to predict the native state for a given protein sequence. This paper attempts to find an efficient heuristic algorithm that can obtain the native state for the given protein sequence with the global energy minimum. As an improvement of particle swarm optimization (PSO) [4] algorithm, the proposed algorithm introduces a new architecture of PSO algorithm, which combines local search method with global search method, attempting to balance exploration and exploitation capability to be more effective in protein folding prediction.

1-4244-1120-3/07/$25.00 ©2007 IEEE

II. IMPROVED PSO ALGORITHM

The PSO algorithm [4] is an evolutionary technique based stochastic optimization paradigm introduced in 1995 by Eberhart and Kennedy. There exist two different kinds of capability in PSO algorithm: exploration capability and exploitation capability. In PSO algorithm, each particle is attracted to a certain position from population best experience and the particle's best experience in order to search the better position. With an increasing of the size of the population participating in searching the better position, the probability of finding the local best solution must increase. The shared information is employed effectively in evolutionary strategy, and it makes the efficiency of exploring in the neighborhood area higher. Nevertheless, when all of particles almost participate in the work of exploitation, the capability of exploration must be influenced negatively and be limited to a low level. Consequently, it is so hard to find a better solution. Therefore, it is essential for PSO algorithm to balance the abilities of exploitation and exploration effectively. In order to solve the problem mentioned above, this paper improves PSO algorithm structure to increase the global search capability of PSO algorithm.

A. Improved PSO Algorithm

The algorithm proposed in this paper is a modified version of PSO algorithm, where the whole population having N particles divides into three subgroups: elitist subgroup, exploitative subgroup, and explorative subgroup. In this new algorithm structure, at time t the elitist subgroup $P_1(t)$ includes n_1 particles; the exploitative subgroup $P_2(t)$ includes n_2 particles; and the explorative subgroup $P_3(t)$ includes n_3 particles ($N = n_1 + n_2 + n_3$). Firstly, the population $P(t)$ at time t is sorted by ascending according to the energy value and a list of population is generated by the population index i ($i \in [1, N]$). Particles for index between 1 and n_1 are grouped into the elitist subgroup $P_1(t)$. Particles for index from ($n_1 + 1$) to ($n_1 + n_2$) are grouped into the exploitative subgroup $P_2(t)$. Besides $P_1(t)$ and $P_2(t)$, the remaining particles form the explorative subgroup $P_3(t)$. And there are two kinds of fundamental strategies used in this new algorithm.

- Mutation Strategy. The Gaussian mutation used in GA can be introduced in the improved PSO algorithm, where each particle moves to next position inside the search area with a predetermined probability without being affected by other individuals. However, the improved PSO algorithm leaves a certain ambiguity in the transition to the next generation due to Gaussian mutation. This strategy employs the following equation:

$$mut(x) = x \cdot (1 + gaussian(\sigma)) \tag{4}$$

Where σ is set to be 0.1 times the length of the search space in one dimension, that is also get from some experiments. x denotes a numerical value for each particle position. These particles are selected at the predetermined probability and their positions are determined at the probability under the Gaussian distribution. Wide-space searches are possible at the initial search stage and mutation rate is decreasing gradually at the middle and final stages. Linearly decreasing this mutation rate starting at 1.0 and ending at 0 is used.

- Exploration Strategy: Exploration is a kind of computational resources allocated for finding better solution. This approach has some advantages because it allows the exploration of new regions in the search space while retaining the ability to improve good solutions already found. The explorative subgroup mainly aims for this object. If a more suitable solution cannot be found in the neighborhood, all of individuals in explorative subgroup will regenerate and search in the new neighborhood until a satisfied one (that is better than the original one) is achieved. The center point of the new neighborhood is the best position the population found so far and the radius constrains a majority of individuals to move in the neighborhood. The formula of updating the explorative subgroup is

$$x\ (t+1) = G(t) + rand() \cdot R(t+1) \tag{5}$$

$G(t)$ denotes the best position found by the entire population. The symbol $R(t+1)$ represents the radius and it can change and adjust adaptively according to the formula

$$R(t+1) = \begin{cases} R(t) \cdot (1+\alpha), & T(t) > T_0 \\ R(t), & T(t) \le T_0 \ \&\ \Delta\varepsilon > \varepsilon_0 \\ R(t) \cdot (1-\alpha), & T(t) < T_0 \ \&\ \Delta\varepsilon \le \varepsilon_0 \end{cases} \tag{6}$$

In (6), $\Delta\varepsilon = \Phi_{best}(t+1) - \Phi_{best}(t)$, ε_0 is a threshold of $\Delta\varepsilon$, and $T(t)$ is a kind of additional counter and T_0 is a threshold of T , α denotes the increasing rate or decreasing rate of the radius.

$$T(t) = \begin{cases} T(t-1) + 1, & \Delta\varepsilon > \varepsilon_0 \\ 0, & \Delta\varepsilon \le \varepsilon_0 \end{cases} \tag{7}$$

B. Description of the Improved PSO Algorithm

As mentioned above, the PSO algorithm is a kind of stochastic optimization algorithm. In order to avoid the premature of the PSO algorithm, this paper proposes a kind of improved PSO algorithm based on toy model. The improved PSO algorithm describes in detail as follows:

Step 1. Initiation. The N particles generated randomly and initially are included into the population $P(t)$ $(t = 1)$. Then begin to set three kinds of factors: elitist factor r_1 ($n_1 = N \cdot r_1$), exploitative factor r_2 ($n_2 = N \cdot r_2$),

explorative factor r_3 ($n_3 = N \cdot r_3$). The size of the population is invariable, so $r_1 + r_2 + r_3 = 1$. Additionally, the initiative value of ε_0, T_0 and α in the formula (6) are set as 0.001, 50 and 0.005, the maximum iterations time $L_{\max}$ is 20000.

Step 2. Energy Values. The energy value $Energy_i$ of each particle is gained according to the formula (1).

Step 3. Sorting. According to energy values obtained in Step2, all the particles are sorted by ascending. In the meantime, if the best energy E_0 is lower than the best accepted solution E_{gbest} found so far, the best energy E_{gbest} will be replaced by the E_0 and the best position is updated as well.

Step 4. Evolution. Three different subgroups evolve respectively according to different approaches. In the beginning, the elitist subgroup $P_1(t+1)$ is inherited directly from last generation. When the number of iteration time is large properly, this subgroup $P_1(t+1)$ is obtained by mutation strategy. For the exploitative subgroup $P_2(t+1)$, the positions of all the particles are updated through the PSO algorithm. Meanwhile, for the explorative subgroup $P_3(t+1)$, the positions of all the particles are generated randomly with the exploration strategy. Consequently, the new population $P(t+1)$ is obtained. $(P(t+1) = P_1(t+1) \bigcup P_2(t+1) \bigcup P_3(t+1))$

Step 5. Terminative requirements. The process terminates if the current iteration time $L+1 > L_{\max}$, and it obtains the best energy E_{gbest} and the best position of it.

Step 6. Active counter L. The counter L adds by 1, and then the process returns to Step2.

III. Experimental Results

In order to evaluate the performance of the improved PSO algorithm used in the prediction of protein folding, we restrict ourselves to toy model with "Fibonacci sequences'' studied in Refs [5][7], where A and B behave respectively as hydrophobic and hydrophilic. Fibonacci sequences are defined recursively by S_0=A, S_1=B, S_{i+1}=S_{i-1}* S_i. Here the asterisk is the concatenation operator. The first few sequences are S_2=AB, S_3=BAB, S_4=ABBAB, etc. Hydrophobic residues A occur isolated along the chain, while hydrophilic residues B occur either isolated or in pairs. For protein sequences that the length of them is less than 13, the minimum energies gained through the improved PSO algorithm agree perfectly with those of Stillinger [5]. Therefore, we consider the protein sequences with length 13, 21, 34, and 55 listed in Table I.

The algorithm has implemented by C++ in WindowsXP. The parameters [7] set in the experiments are as the following value: the maximum number of iterations $L_{\max}$ =20000; the population scale N =1000; the elitist factor r_1 =0.05; the exploitative factor r_2 =0.7; the explorative factor r_3 =0.25.

TABLE I. THE FOUR FIBONACCI SEQUENCES AND MINIMUM ENERGIES OF THESE SEQUENCES

L	E_{HTML}	E_{PERM}	E_{nPERM}	E_{PSO*}
13	-3.2235	-3.2167	-3.2939	-3.2941
21	-5.2281	-5.7501	-6.1976	-6.1977
34	-8.9749	-9.2195	-10.7001	-10.7036
55	-14.4089	-14.9050	-18.5154	-18.4236

Table I shows the predicted energy minima for all the four Fibonacci sequences based on the toy model obtained by the PSO algorithm (E_{PSO*}), compared with the results obtained by the high temperature Monte Carlo method (E_{HTML}) [5], pruned enriched Rosenbluth method (E_{PERM}) [6], and the improved pruned enriched Rosenbluth method (E_{nPERM}) [6]. The experimental results show that our results are better than those of the HTMC and the PERM for the four sequences, with the minimum energy difference increasing gradually for longer chains. Compared with results of nPERM, for sequences with length 13, 21 and 34, our predicted results are also slightly better than them. However, for the sequence with length 55, we can not reach the result yielded by nPERM. It puts emphasis on that the nPERM is split into two independent processes. In the first process, it employs PERM to calculate the energy minimum of protein sequence with toy model; meanwhile it records the optimal solution with the energy minimum. In the second process, the solution obtained by the first process is used for initializing the primal solution, and then it adopts the subsequent conjugate-gradient minimization to predict the minimum of protein sequence. So the result gained by nPERM is also considered as the best one. Compared with nPERM, the improved PSO algorithm only has one process to predict protein folding. Consequently, the performance of the improved PSO algorithm is better than that of nPERM in the aspect of computational efficiency to some extent. It shows that the improved PSO algorithm is tuned to give good results of protein folding prediction.

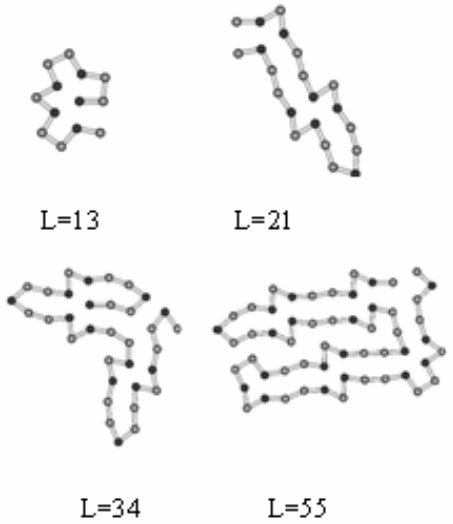

Figure 1. The lowest-energy conformations of the two-dimensional toy model for test sequences. The black dot indicates hydrophobic A monomers and the gray dot indicates hydrophilic B monomers

Figure 1 depicts four graphs of the global energy minimum conformation of the four protein sequences in two-dimension by the improved PSO algorithm. It is clear that each conformation forms a single hydrophobic core for the test sequence. In other words, the hydrophobic residues are always flanked by hydrophilic residues along the chain. It should be noted that the hydrophobic core in the conformation of the sequence with length 13 is more compact than that in other conformations. This indicates that although the toy model reflects the real protein folding structure in two-dimension to some extent, it is still not extremely perfect.

IV. CONCLUSION

This paper describes an improved PSO algorithm and its application in protein folding prediction based with toy model. The experimental results show that the improved PSO algorithm appears to have the ability to search global optimal solution in consecutive space. The mutation strategy and exploration strategy used in the algorithm help to escape from the local minimum. As one of our future work, we would like to expand the improved PSO algorithm to handle the three-dimensional protein folding problem.

ACKNOWLEDGMENT

The authors thank all the anonymous referees for their helpful comments. They also thank Xiao-Li Lin and Jin Lu for improving the presentation

REFERENCES

[1] K. A. Dill, "Theory for the folding and stability of globular proteins," *Biochemistry*, vol. 9, pp. 1501-1512, 1985.

[2] F. H. Stillinger, T. H Gordon, and C. L. Hirshfeld, "Toy model for protein folding," Physical Review E, vol. 48, pp. 1469-1477,1993.

[3] P. Crescenzi, et al, "On the complexity of protein folding," Proceedings of the 2nd Conference on Computational Molecular Biology, 1998.

[4] J. Kennedy, and R. C. Eberhart, "Particle Swarm Optimization," Proceedings of the IEEE Intl. Conf. Neural Networks, pp.1942-1948, 1995.

[5] F. H. Stillinger, "Collective aspects of protein folding illustrated by a toy model," Physical Review E, vol. 52, pp. 2872-2877, 1995.

[6] H. P. Hsu, V. Mehra, and P. Grassberger, "Structure optimization in an off-lattice protein model," Physical Review E, vol. 68, 2003.

[7] Y. Shi, and R. Eberhart., "A modified particle swarm optimizer," IEEE Intl. Conf. on Evolutionary, pp. 69-73, 1998.