Engineering Applications of Artificial Intelligence 27 (2014) 70–79

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Protein secondary structure optimization using an improved artificial bee colony algorithm based on AB off-lattice model

Bai Li [a], Ya Li [b,*], Ligang Gong [c]

[a] School of Advanced Engineering, Beihang University, Beijing 100191, China
[b] School of Mathematics and Systems Science & LMIB, Beihang University, Beijing 100191, China
[c] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

ARTICLE INFO

Article history:
Received 15 May 2013
Accepted 14 June 2013
Available online 18 July 2013

Keywords:
Artificial Bee Colony algorithm (ABC)
AB off-lattice model
Protein secondary structure optimization
Convergence of algorithm

ABSTRACT

Predicting the secondary structure of protein has been the focus of scientific research for decades, but it remains to be a challenge in bioinformatics due to the increasing computation complexity. In this paper, AB off-lattice model is introduced to transforms the prediction task into a numerical optimization problem. Artificial Bee Colony algorithm (ABC) is an effective swarm intelligence algorithm, which works well in exploration but poor at exploitation. To improve the convergence performance of ABC, a novel internal feedback strategy based ABC (IF-ABC) is proposed. In this strategy, internal states are fully used in each of the iterations to guide subsequent searching process, and to balance local exploration with global exploitation. We provide the mechanism together with the convergence proof of the modified algorithm. Simulations are conducted on artificial Fibonacci sequences and real sequences in the database of Protein Data Bank (PDB). The analysis implies that IF-ABC is more effective to improve convergence rate than ABC, and can be employed for this specific protein structure prediction issues.

1. Introduction

The Central Dogma of genetics demonstrates how genetic information flows from DNA to mRNA and then to amino acids. However, the mechanism of direct translation from amino acids to functional structure has not been thoroughly revealed after decades of intensive work (Gozuacik and Adi, 2004). On the way to construct a functional structure, protein secondary structure folding plays an essential role, which is valuable in determining sub-cellular locations and improving the sensitivity of fold recognition methods (Montgomerie et al., 2006). Therefore, protein secondary structure prediction has become a routine part of protein analyses and annotation in recent years (Pollastri et al., 2007). X-ray diffraction analysis method and nuclear magnetic resonance method (NMR) are two widely used methods to obtain the secondary conformation of protein (Chiu et al., 2005). Most of the known conformations are obtained by them and stored in the Protein Data Bank (PDB) database (Murzin et al., 1995). However, because of the strict experimental condition requirement, researchers are motivated to shift their interest towards the establishment of effective models for simulation (Kim et al., 2005).

There are many models or methods intended for protein secondary structure prediction. For example, Jpred (Cole et al., 2008), PSI-PRED (McGuffin et al., 2000), and PHD-PSI (Przybylski and Rost, 2002) are methods based on neural networks. Context-based secondary structure potential approach (CSSP) (Li et al., 2013), and some other statistical classification methods, such as *K*-nearest neighbor method (Joo et al., 2004), support vector machines (SVM) (Zhou et al., 2008), and hidden Markov model (Malekpour et al., 2009) have been well investigated. Since Christian Anfinsen proposed that the native states of proteins reside in the free-energy minima (Anfinsen, 1973), the thermodynamic hypothesis has been widely accepted as a new "Central Dogma" in the field of protein folding. Stillinger et al. (1993) established AB off-lattice model, where each amino acid is treated as a hydrophobic or hydrophilic particle. Particles are linked up by chemical bonds, which are unbendable but free to rotate. In this model, particles prefer locations with corresponding potential energy values as low as possible. In this way, protein secondary structure folding process is transformed into a numerical optimization problem. Unfortunately, it is not easy to enumerate global optimum in this way, especially in terms of high-dimension cases, which is considered to be a NP-complete problem (Berger and Leighton, 1998), i.e., unsolvable in polynomial time.

To avoid an inefficient enumerating process, researchers investigated evolutionary algorithms such as genetic algorithm (GA) (Holland, 1992), and differential evolution algorithm (DE) (Storn and Price, 1995). GA was initially proposed to simulate the

* Corresponding author. Tel.: +86 15801211607.
E-mail addresses: libaioutstanding@163.com (B. Li), yli@buaa.edu.cn, liya414@yahoo.com (Y. Li), glg@aspe.buaa.edu.cn (L. Gong).

0952-1976/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2013.06.010

self-adaptation behavior of natural systems. Guided by the selection and crossover process for self-adaption, GA is capable of global search and has strong robustness, even without any given knowledge of the system. The major difference of DE from GA is its selection operation for crossover and mutation, which contribute to higher convergence rate and thus improve the poor local searching ability of GA. Swarm intelligence algorithms are another focus of research. Ant colony optimization algorithm (ACO) (Clark, 2007), particle swarm optimization algorithm (PSO) (Kennedy and Eberhart, 1995), and artificial bee colony algorithm (ABC) (Karaboga, 2005) are three typical examples. Inspired by the social behavior of bird flocking, PSO gives consideration to both local and global search abilities and is imperfect in its ability to overcome premature convergence. ABC is motivated by the foraging behavior of bee swarms, in which both local exploitation and global exploration are conducted in iterations. It works well in global exploration but is poor in the exploitation process. Many improvements have been made for ABC in different ways. Manuel and Elias (2013) adopt a modified ABC for FIR filter. In Xiang and An (2012), an efficient and robust ABC (ER-ABC) is proposed. Gao and Liu (2011) introduce Rosenbrock's rotational direction method to revise ABC for accurate numerical optimization. Vector-evaluated strategy is implemented in VE-ABC (Omkar et al., 2011). In Kang et al. (2013), the initial population of bee swarm is produced by both chaotic theory and opposition-based learning method. Viewing all those improvements made for ABC, we find that few researchers have paid adequate attention to the utilization of previous convergence states as feedback information to guide subsequent searching process. Moreover, as stated in our earlier work (Li and Li, 2012), the balance between local exploitation and global exploration should not be ignored during the iterations when using ABC. Therefore, we propose a novel internal feedback information based artificial bee colony algorithm (IF-ABC) for better converging performance. Internal feedback strategy (IFS) mainly works to reflect the states of convergence performance and then to guide the subsequent searching process. By designing such a self-adaptive system, fewer user-specified parameters or initial values are set in IF-ABC compared with other evolutionary algorithms.

Many algorithms have been applied on AB off-lattice model for the specific protein secondary structure optimization problems (Hsu et al., 2003; Kim et al., 2005; Shmygelska and Hoos, 2005; Zhang and Lin, 2006; Liu et al., 2005, 2010; Wang and Zhang, 2009; Cutello et al., 2007; Lin and Zhu, 2008). However, previous work has seldom rigorously considered the accuracy of this off-lattice model and the effectiveness of the optimization results. In other words, it is still not clear if the reported "optimal" structures are indeed the ground states in the seemingly complicated energy landscape. Therefore, we conduct simulations and compare the result of IF-ABC with those of some other algorithms. And some in-depth analyses are also made regarding the similarity of such optimized conformations and real structures in the PDB database.

The remainder of this paper is organized as follows. In Sections 2 and 3, basic principle of AB off-lattice model and artificial bee colony algorithm are given. Section 4 is an introduction to the mechanism of IF-ABC. In Section 5, we conduct simulations and release the optimal results. In this section, comparisons are made on both artificial Fibonacci sequences and real sequences. In Section 6, we make a generalization of the performance of IF-ABC. The complete convergence proof of the algorithm is provided in Appendix.

2. Principle of AB off-lattice model

AB off-lattice model, also known as toy model in bioinformatics, has been widely used to describe the protein secondary structure folding process for decades (Stillinger et al., 1993). It is based upon the viewpoint that the native structure of a protein corresponds to the very structure among possible ones with the lowest free energy value. This energy consists of two parts, one is the intermolecular interaction among protein atoms, and the other is the intermolecular interaction between proteins and surrounding solvent molecules (Anfinsen, 1973).

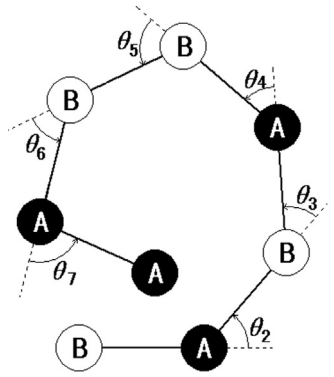

Fig. 1. Schematic diagram of AB off-lattice model on a 2D surface.

In this model, 20 kinds of amino acids (basic building blocks of proteins) are classified into two categories, named hydrophobic residues and hydrophilic residues. Fig. 1 shows the schematic diagram of this model in 2D surface, where hydrophobic residues and hydrophilic residues are represented by A and B particles, respectively. Particles are linked up by chemical bonds and thus form a non-directional chain. The conformation of any chain with n particles is specified by the $(n-2)$ bend angles $[\theta_2, \theta_3, \ldots, \theta_7]$ as shown in Fig. 1. It is arbitrarily set that $\theta_i \in [-180^\circ, 180^\circ)$ for each bend angle in this model. Obviously, $\theta_i \in [-180^\circ, 0^\circ)$ means a counterclockwise rotation trend in the chain, and $\theta_i \in (0^\circ, 180^\circ)$ indicates a clockwise rotation trend.

The free-energy function *Energy* of a sequence of amino acids is defined by

$$Energy = \sum_{i=2}^{n-1} \frac{1-\cos\theta_i}{4} + 4 \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} [r_{ij}^{-12} - C(\xi_i, \xi_j) r_{ij}^{-6}]. \quad (1)$$

The property of the ith individual particle is reflected by ξ_i. If residue i is hydrophilic, then $\xi_i = 1$; otherwise, $\xi_i = -1$. r_{ij} denotes the distance between particles i and j in the chain

$$r_{ij} = \sqrt{\left[1 + \sum_{k=i+1}^{j-1} \cos\left(\sum_{l=i+1}^{k} \theta_l\right)\right]^2 + \left[\sum_{k=i+1}^{j-1} \sin\left(\sum_{l=i+1}^{k} \theta_l\right)\right]^2} \quad (2)$$

$C(\xi_i, \xi_j)$ represents the interaction between two particles:

$$C(\xi_i, \xi_j) = \tfrac{1}{8}(1 + \xi_i + \xi_j + 5\xi_i\xi_j). \quad (3)$$

It is easy to show that, coefficient $C(\xi_i, \xi_j)$ equals to 1 for AA pairs, 0.5 for BB pairs, and -0.5 for AB or BA pairs. It is based on the assumption that correlations between AA particles should be strongly enhanced, while BB particles are weakly encouraged; otherwise it results in a weak repulsion.

In this way, the protein secondary structure prediction task is transformed into a numerical optimization problem through AB off-lattice model. Fig. 2 sketchily depicts a wireframe parametric surface as well as the corresponding contour plot for a sequence of ABBA. Note that the ground state is approximately -0.036, and the corresponding global optimal solution $[-86.22, -85.40]$ is located in a narrow "valley", which is not easy to be obtained.

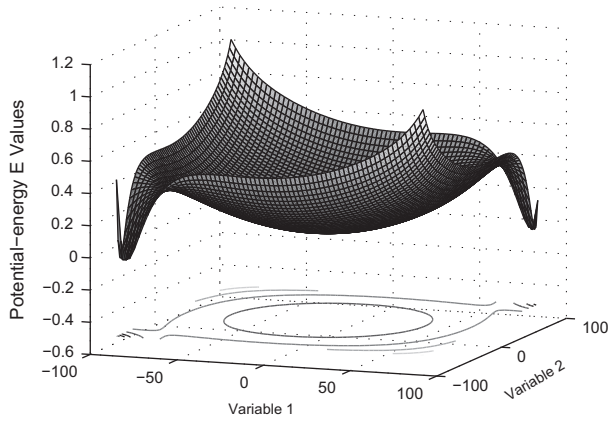

Fig. 2. Mesh plot of potential energy values for sequence ABBA in AB off-lattice model.

3. Principle of artificial bee colony algorithm

The algorithm of ABC implements a process of iterative optimization, during which the bee colony consists of three groups: employed bees, onlooker bees, and scout bees (Karaboga, 2005). Let $\mathbf{X}_i=(X_i^1,X_i^2,\ldots,X_i^D)$ $(i=1,\ldots,SN)$ represent a position in the food source searched by the ith employed bee or its corresponding onlooker bee. Here, a position implies a possible solution to the optimization problem and obviously, the number of employed bees is SN. Onlooker bees work to search locally around their corresponding employed bees, which means that the number of onlookers is also SN. Any employed bee and its corresponding onlooker bee who cannot find any better position in a certain number of iterations will be replaced by a scout bee.

At first, all the employed bees set out to explore randomly in the searching space (i.e., the feasible solution space). Particularly, the process to initialize the jth element of the ith solution $\mathbf{X}_i$ is described as below

$$X_i^j \leftarrow X_{min}^j + rand(0,1)(X_{max}^j - X_{min}^j), \quad j=1,2,\ldots,D, \tag{4}$$

where X_{min}^j and X_{max}^j denote the lower and upper boundaries of this jth element, and D denotes the dimension of any $\mathbf{X}_i$.

In each of the iterations, an employed bee executes a crossover and mutation process to share information with one randomly chosen companion and search in the new position as follows:

$$X_i^j \leftarrow X_i^j + rand(0,1)(X_k^j - X_i^j) \tag{5}$$

In this equation, the ith employed bee exchanges information with the kth one in its jth element. It is noted that only one element is changed for each of the employed bees during this process. Afterwards, greedy selection strategy is implemented. If the refreshed position is better (i.e. the corresponding free-energy value is lower), the previous position is abandoned; otherwise, the employed bee remains at its previous position and the crossover process mentioned above is of no avail.

Then, each of the onlookers randomly chooses to exploit or not around its corresponding employed bee's position with probability defined as follows:

$$Pi = \frac{obj(\mathbf{X}_i)}{\sum_{j=1}^{SN} obj(\mathbf{X}_j)}, \tag{6}$$

where $obj(\cdot)$ denotes the objective function value. It is obvious that higher $obj(\mathbf{X}_i)$ enjoys higher probability of being selected by the corresponding onlooker bee. Such approach is known as the famous *roulette selection strategy*.

These onlooker bees search around corresponding employed bees by Eq. (5) again. Here, kth companion is still randomly selected. Afterwards, greedy selection strategy is applied on onlooker bees in a similar way.

For each of the employed bees, together with the corresponding onlooker bee, we let the parameter *trial* represent the number of inefficient searching iterations before better position is derived. If the ith employed bee or the ith onlooker bee finds a better position, $trial(i)$ is set to zero; otherwise, it is added by one for the next iteration. A prior parameter *limit* is arbitrarily set as a threshold. If $trial(i) \geq limit$, the current ith position for the ith employed bee to explore or the ith onlooker bee to exploit should be abandoned. Meanwhile, a scout bee takes this place with randomly initialized position by Eq. (4).

The pseudo-code of ABC for constrained optimization problems is given below. Note that *MCN* refers to the maximum iteration number.

```
1   Initialize solution population using Eq. (4)
2   Set iter = 1
3   repeat
4       while iter ≤ MCN, do
5           generate positions for employed bees by Eq. (5)
6           evaluate and greedily select employed bees
7           if position is improved, do
8               trial(i) ← 0
9           end if
10          calculate P_i by Eq. (6)
11          if P_i > rand(0,1), do
12              generate positions for onlooker bees by Eq. (5)
13              evaluate and greedily select onlookers
14              if position is improved, do
15                  trial(i) ← 0
16              else
17                  trial(i) ← trial(i) + 1
18              end if
19          end if
20          if trial(i) ≥ limit, do
21              initialize the position by Eq. (4)
22          end if
23          record current best solution
24          iter ← iter + 1
25      end while
26  end repeat
27  output global optimum
```

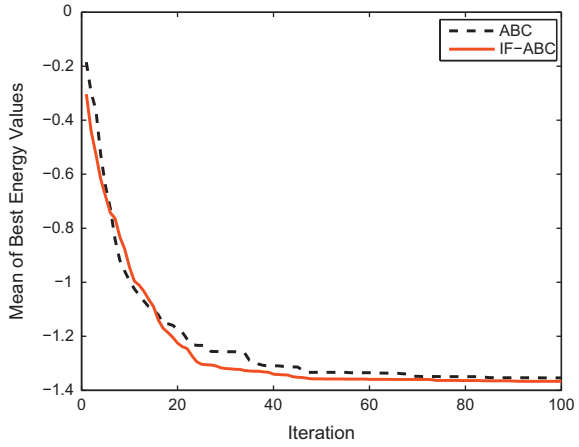

Fig. 3. Mean of best free-energy values for sequence ABAAB ($N=5$).

4. Principle of internal feedback based artificial bee colony algorithm

The algorithm of IF-ABC mainly differs from original ABC in the utility of internal feedback information *trial*(*i*) and in the removal of roulette selection strategy.

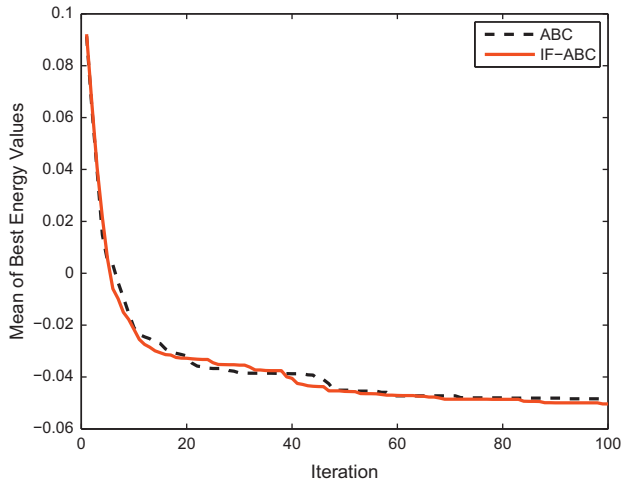

Fig. 4. Mean of best free-energy values for sequence ABBBB ($N=5$).

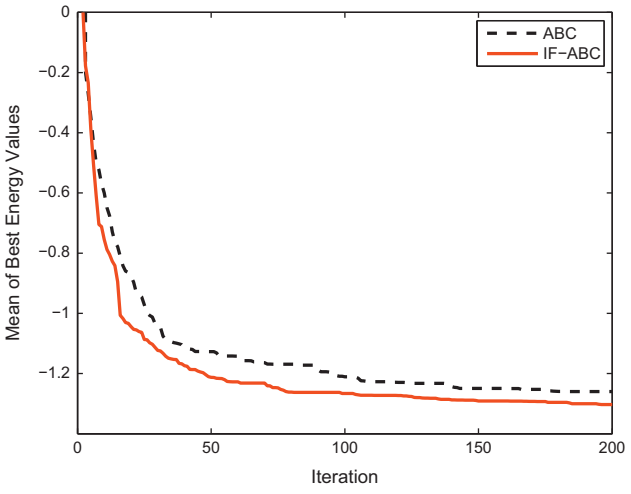

Fig. 5. Mean of best free-energy for sequence AABABB ($N=6$).

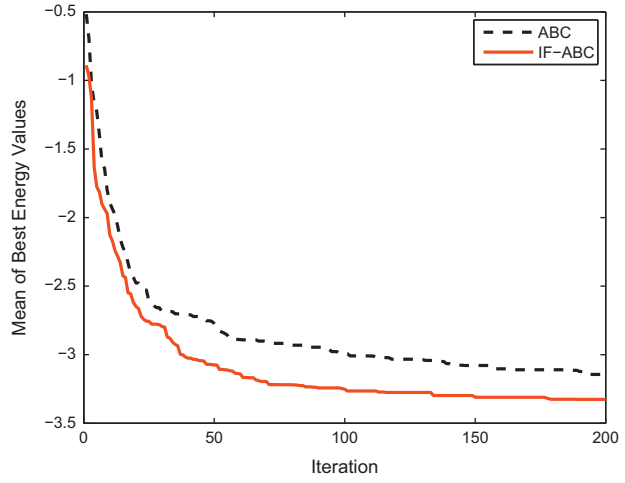

Fig. 6. Mean of best free-energy values for sequence AAABAA ($N=6$).

Notice that the adoption of roulette selection strategy and the employment of scout bees are the two primary factors to affect the convergence performance (Li and Li, 2012) of ABC. Scout bees are employed to avoid getting trapped in local optimums by the initialization of their positions, which is not adequately efficient for convergence performance in subsequent iterations. Roulette selection strategy works to evaluate current corresponding positions of employed bees (see Eq. (6)). This strategy will greatly guide the subsequence convergence performance and thus is crucial. In fact, for some engineering optimization problems, it is inevitable to be trapped in local optimums. The situation could be even worse if the local optimum found is significantly superior, which is often regarded as "super individual" in a swarm. Under such circumstances, *Pi* of any other employed bee is much lower, which prevents the corresponding onlooker bee to follow it.

However, the searching competence of an employed bee should not be evaluated by the quality of its current position (i.e. the objective function value), but by the efficiency of current search, i.e., by *trial*. Although roulette selection strategy significantly contributes to improve the convergence speed, it should be noted that the primary concern is to avoid prematurely trapped in local optimums for some long-term numerical optimizations. In conclusion, we replace roulette selection strategy by the parameter *trial* to reveal internal convergence states so as to avoid premature convergence.

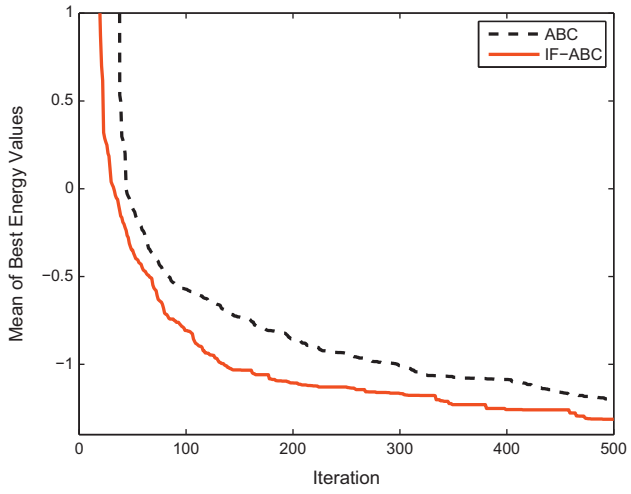

Fig. 7. Mean of best free-energy values for sequence ABBABBABABBAB ($N=13$).

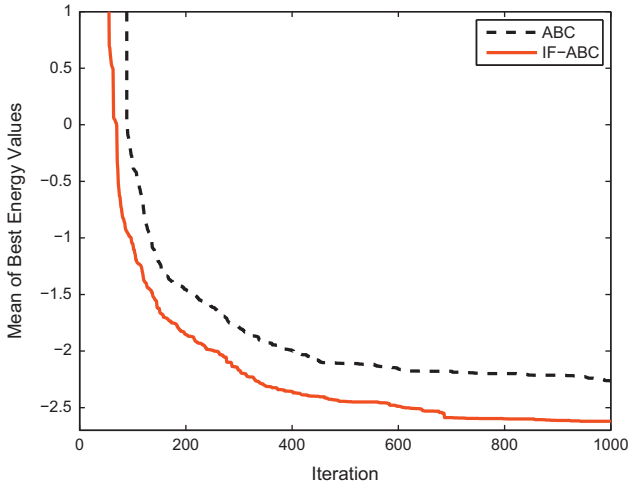

Fig. 8. Mean of best free-energy values for sequence BABABBABABBABBABABBAB ($N=21$).

In the crossover and mutation process of IF-ABC, the number of coordinates to change in $\mathbf{X}_i$ is considered to be a variable, but not fixed to be 1. Eq. (5) is implemented on employed bees for $trial(i)$ times, where $1 \le trial(i) \le D$ is an integer. Same as ABC, for the ith employed bee and its corresponding onlooker, if neither of them finds a better solution, then $trial(i)$ is added by 1; otherwise, $trial(i)$ is reset to be 1 (but not 0). If $trial(i)$ is greater than D, then the current ith position $\mathbf{X}_i$ should be initialized by Eq. (4).

For each of the onlooker bees, only one coordinate should be changed during the exploitation process, because multi-crossover process contributes little to local searching ability. The exploitation ability for onlooker bees can be enhanced by introducing a convergence factor γ in the crossover process,

$$X_i^j \leftarrow X_i^j + \gamma(i) rand(0,1)(X_k^j - X_i^j), \tag{7}$$

where

$$\gamma(i) = \exp\left\{-[trial(i)-1]\frac{\ln 10}{D-1}\right\} \tag{8}$$

As shown in Eq. (8), $\gamma(i)$ decreases exponentially to 0.1 as $trial(i)$ approaches D. Remark that this boundary can be chosen as any real number if we modify Eq. (8). But for the onlooker bee with large $trial(i)$, it is assumed that the locally exploitation process should be intensified for improvement. In particular, when $trial(i) = 1$, we have $\gamma(i) = 1$, as is the case in original ABC.

To briefly conclude, *trial* in IF-ABC works to manipulate the scales of local exploitation accuracy and to guide crossover and mutation behavior in global exploration process.

The pseudo-code of IF-ABC for numerical optimization is given below. Again, let *MCN* represent the maximum iteration number.

```
1.   Initialize solution population by Eq. (4)
2.   Set any trial(i) = 1, and iter = 1
3.   repeat
4.       while iter ≤ MCN, do
5.         for k = 1 : trial(i), do
6.             crossover and mutate by Eq. (5) for X_i^j
7.   end for
8.   generate position for each employed bee
9.   evaluate fitness values and adopt greedy selection
10.  if ith position is improved, do
11.        trial(i)←1
12.  end if
13.        generate positions for onlooker bees by Eq. (5)
14         evaluate and greedily select onlookers
15         if position is improved, do
16     trial(i)←1
17     else
18     trial(i)←trial(i) + 1
19         end if
20     end if
21     if trial(i) > D, do
22         initialize ith position by Eq. (4)
23     end if
24     memorize current best solution
25     iter←iter + 1
26         end while
27     end repeat
28     output global optimum
```

The convergence proof of IF-ABC is given in Appendix.

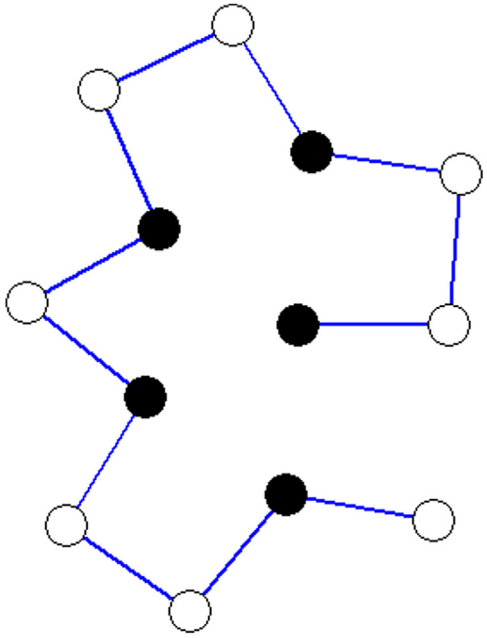

Fig. 9. The best optimized conformations for Fibonacci sequence ($N = 13$) obtained by IF-ABC. The optimal solution $\mathbf{X}$ is {85.4499, 85.9482, −49.7703, 84.7718, 87.0443, −83.9978, 111.9129, −83.0073, 86.8928, 84.9553, −59.6743}.

Table 1
Mean free-energy values of artificial sequences calculated by ABC and IF-ABC.

Artificial Sequences	N	*Energy* derived by ABC (value of *SN*/number of iterations)			*Energy* derived by IF-ABC (value of *SN*/number of iterations)		
		20/1000	40/1000	60/1000	20/1000	40/1000	60/1000
ABAAB	5	−1.3666	−1.3752	−1.3762	−1.3746	−1.3763	−1.3765
ABBBB	5	−0.0610	−0.0643	−0.0651	−0.0639	−0.0652	−0.0654
AABABB	6	−1.2991	−1.3176	−1.3332	−1.3134	−1.3298	−1.3473
AAABAA	6	−3.5287	−3.4779	−3.5621	−3.4610	−3.5455	−3.6230
ABBABBABABBAB	13	−1.3135	−1.4620	−1.4942	−1.4200	−1.6735	−1.7570
BABABBABABBABBABABBAB	21	−2.3600	−2.4369	−2.5723	−2.5723	−2.6946	−2.8991

Table 2
Minimal energy values of artificial sequences obtained by algorithms Pruned Enriched Rosenbluth Method (PERM), Genetic-Annealing Algorithm (GAA), Evalutionary Particle Swarm Optimization (E-PSO), improved Tabu Search Algorithm (I-TS) and IF-ABC.

Artificial Sequences	N	PERM	GAA	E-PSO	I-TS	IF-ABC
ABBABBABABBAB	13	−3.2167	−3.2940	−3.2941	−3.2941	−3.2941
BABABBABABBABBABABBAB	21	−5.7501	−6.1838	−6.1980	−6.1953	−6.1980
ABBABBABABBABBABABBABABBABBABABBAB	34	−9.2195	−10.5992	−9.8341	−10.7070	−10.8060

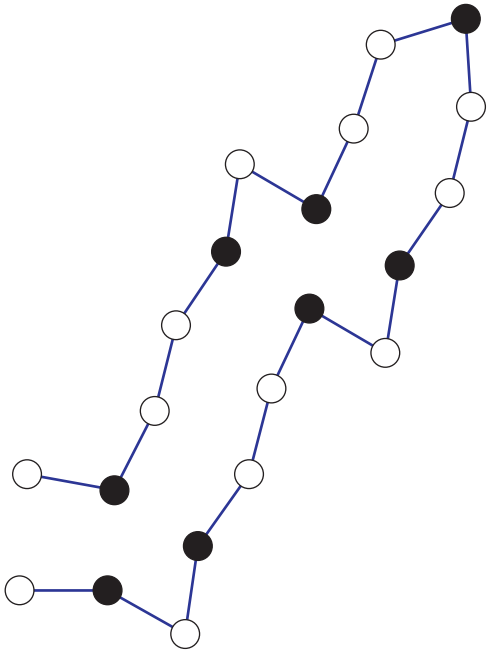

Fig. 10. The best optimized conformations for Fibonacci sequence ($N=21$) obtained by IF-ABC. The optimal solution **X** is {−29.4339, 111.0289, −26.9311, 20.8178, −10.7849, −95.0706, 111.3693, −25.7034, 20.7508, 17.5707, 103.5723, 54.6928, −6.8404, −95.3601, 111.3307, −25.0472, 19.8815, −12.8529, −73.1896}.

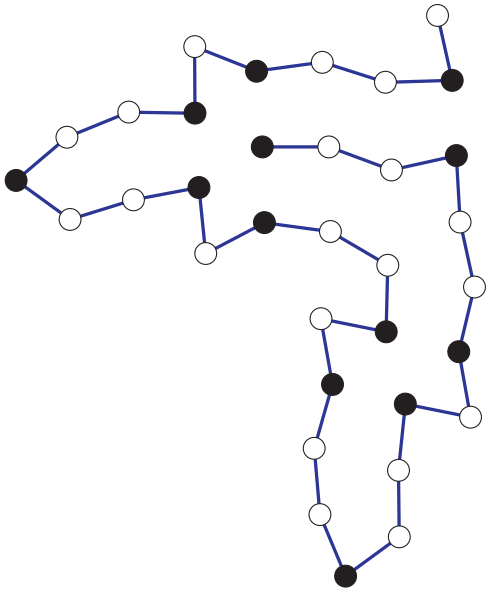

Fig. 11. The best optimized conformations for Fibonacci sequence ($N=34$) obtained by IF-ABC. The optimal solution **X** is {−20.3009, 32.7677, −99.1187, 8.9153, −25.6173, 23.3967, −111.4206, 95.4252, 6.6239, −54.4829, −103.5475, −17.8249, −20.5841, 25.7852, −111.2621, 99.8364, 60.9015, 22.8205, 35.5087, −112.0200, 94.7888, 6.6231, −53.2306, −103.5627, −18.3558, −22.9026, 91.2604, −111.9488, 29.3274, −25.5453, 19.4017, 100.9575}.

Table 3
Minimal energies of real sequences obtained by different algorithms.

ID in PDB	*N*	Current best *Energy*	*Energy* by IF-ABC
1AGT	38	−19.6169 (Liu et al., 2005)	−21.4242
1AHO	64	−21.0853 (Zhou and Han,2010)	−21.1740
2EWH	98	−71.2849 (Pu, 2011)	−71.4336
2YUX	120	−44.3218 (Pu, 2011)	−44.4124

5. Experimental results and discussions

In this section, artificial Fibonacci sequences and natural sequences are optimized to evaluate the performance of IF-ABC. All simulations are implemented in MATLAB R2010a and executed on an Intel Core 2 Due CPU with 2 GB RAM running at 2.53 GHz. Each of these independent experiments is repeated 30 times with randomly initialized condition.

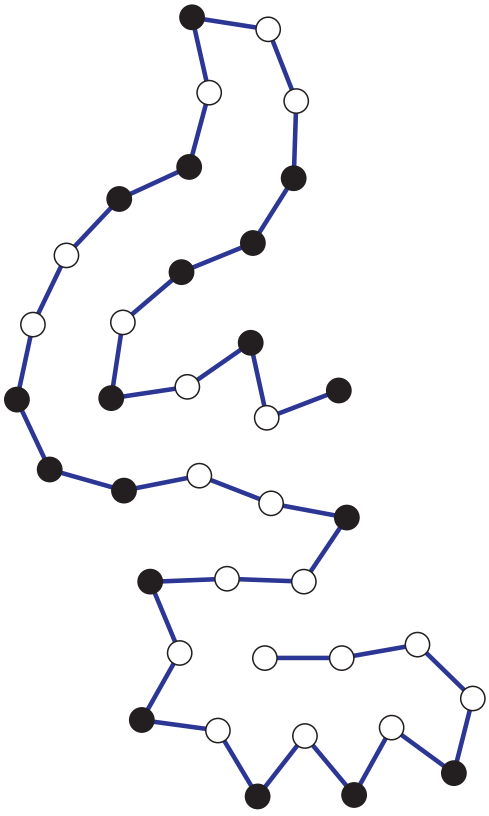

Fig. 12. The best optimized conformations for real sequence 1AGT ($N=38$) obtained by IF−ABC. The optimal solution **X** is {10.0916, −54.5696, −59.6367, −111.9667, 96.8956, −111.0667, 102.3737, −111.3657, 51.4997, −111.6428, 52.0523, −110.4207, −4.3237, 58.2475, 113.2335, −10.6438, 32.4196, −27.0605, −48.9497, −37.1731, −13.6479, −17.7006, −21.9329, 50.3850, 27.5618, −111.5683, −59.7295, −23.1472, −30.4261, −35.7326, 18.5813, 40.6059, 107.2235, 26.4475, −112.5491, 98.7160}.

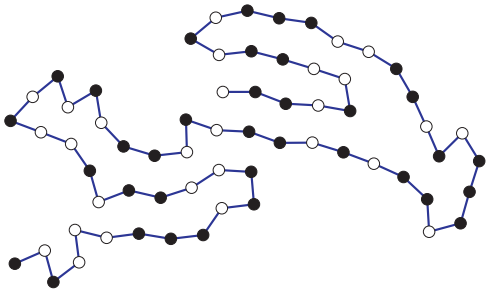

Fig. 13. The best optimized conformations for real sequence 1AHO ($N=64$) obtained by IF-ABC. The optimal solution **X** is {−21.0738, 17.9175, −6.8557, 109.7314, 62.0309, 1.4798, 2.0054, 21.2303, −36.3457, −110.2031, −27.3009, −26.2134, 5.5253, −27.2675, 17.2650, −13.6752, −28.5866, −4.5926, −0.1481, 111.5534, −102.9188, −51.2313, 2.4086, −59.1483, −101.7800, 44.5400, 18.0381, 3.7084, 2.6881, 17.6246, −20.3906, 17.1810, −16.1615, 110.6479, −85.1611, −23.5123, −29.4387, −33.9626, 111.1790, −102.2394, 111.1252, 7.8818, 112.0743, −0.3863, −34.4345, −19.0841, 95.4516, −34.3965, 31.0946, 15.3522, −36.6374, −81.8591, −87.7486, 48.4378, −48.5791, −16.1868, 16.7805, −21.4148, 111.3969, −60.0425, −111.9448, 98.2322}.

For artificial Fibonacci sequences, we let N be the number of amino acids in the secondary structure, and run experiments for the case $N=5$, $N=6$, $N=13$, and $N=21$. Obviously, the dimension $D=N-2$ for these experiments. We compare the convergence curves between IF-ABC and ABC under the condition that $SN=15$ (see Figs. 3–8). Moreover, we also calculate the mean energy values of those sequences with $SN=10$, 20 and 30 respectively. The results derived by ABC and IF-ABC are listed in Table 1 for comparison.

Figs. 3 and 4 show that convergence curves of IF-ABC only differ from those by ABC slightly. However, the convergence accuracy and efficiency of ABC become lower as N increases. It indicates that the advantages of IF-ABC in convergence accuracy gradually emerge as dimensions of sequences increase. By Table 1, we observe that our algorithm has better performance than the original ABC. Larger swarm population results in better convergence performance in most cases listed in the above table, but this is not the fact in the case when $N=21$ and IF-ABC is applied. Therefore, the mechanism that how the swarm population number affects the whole convergence calls for further investigation.

We also run simulations for artificial sequences with 13, 21, and 34 acids. Table 2 compares the free-energy values obtained by nPERM, GAA, E-PSO, I-TS and IF-ABC. It should be noted that longer sequences require a considerable amount of time to optimize. It is obvious that IF-ABC is superior than other algorithms. Figs. 9–11 depict the conformations optimized by IF-ABC, where the black dots represent the hydrophobic A monomers, and the white dots denote hydrophobic B monomers. In Fig. 9, the conformation contains a single hydrophobic core, which reveals the essential characteristic of true protein conformation. When the dimension increases, as in Figs. 10 and 11, although more than one cluster of particles is observed, hydrophobic monomers do show the tendency to converge.

As for real sequences, their dimensions are even larger. We conduct simulations on the sequences named 1AGT, 1AHO, 2EWH, and 2YUX in the PDB database. Table 3 lists the optimal values of those sequences derived from previous literatures, together with the free-energy values obtained by IF-ABC. By comparing those

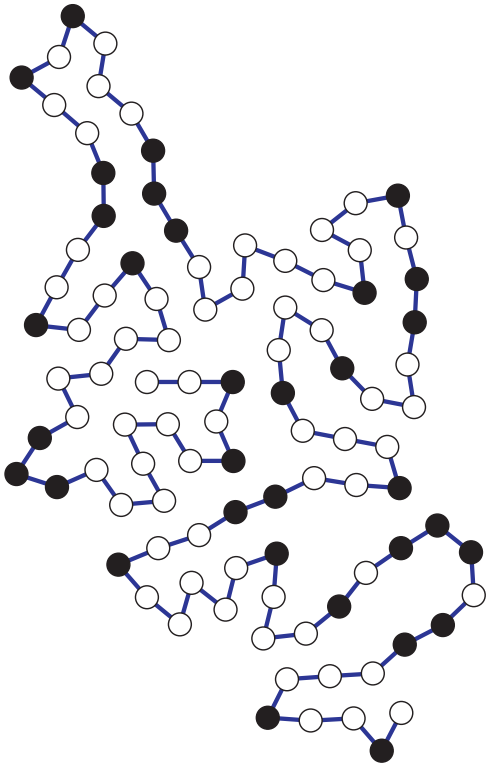

Fig. 14. The best optimized conformations for real sequence 2EWH ($N=96$) obtained by IF-ABC. The optimal solution **X** is {−2.2870, −112.7048, 47.1714, −113.3001, −59.1516, 58.7639, 114.9493, 3.3382, −114.6324, −59.3279, 82.9046, −41.4610, −106.5662, −32.6711, 88.5591, −108.2538, 51.0054, −56.8081, 107.3040, 18.6292, 105.0950, 2.9405, −59.9144, −111.9729, 1.2215, −10.0178, 38.2268, 20.0198, 27.4458, 0.6474, −111.5571, 43.9059, −111.9751, −59.8120, 59.4857, −19.9613, −28.1983, 30.9527, 1.8522, −25.7076, 111.1265, 57.1171, −106.6800, −6.6804, 9.3760, 113.6060, 55.4277, −113.4627, −28.7289, −88.6120, 3.1544, −19.0457, −3.4243, 16.6604, −108.5572, −37.8622, −14.1464, 31.0008, 113.1213, 15.3665, 16.3385, 55.8288, 0.3285, −63.5454, −110.2642, −5.0153, 35.4399, −5.4439, 9.8425, −12.4439, 2.3918, 109.8925, 9.0032, 113.2387, −111.8636, 114.0736, −56.2978, −113.4833, −9.4907, 110.3288, 34.2770, 12.4126, −18.4526, −4.9863, −68.2139, −46.2448, −53.1639, −15.0586, 16.0062, −38.9422, 0.6528, 59.2314, 112.7280, 7.0463, −52.5257, 112.0106}.

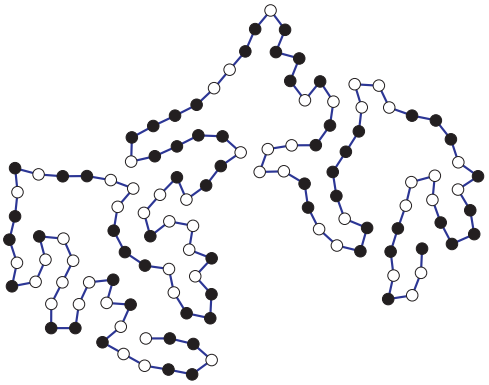

Fig. 15. The best optimized conformations for real sequence 2YUX ($N=118$) obtained by IF-ABC. The optimal solution **X** is {−12.1885, −29.0418, −103.2927, −44.8289, 0.1699, −20.6497, 0.4718, −110.9185, 26.4634, 110.1594, −102.3148, 112.2704, 58.4454, 16.9682, −81.8569, −90.0882, −26.6978, 0.2354, 50.4080, 61.1890, 110.1547, −39.4753, −54.9153, −112.8911, 30.6620, −32.4480, 7.3701, 13.5311, −110.8765, 11.2035, 2.8914, −11.7612, −8.8640, −110.2987, 30.6126, 38.1505, 27.0944, 27.6734, −70.3021, 19.3139, 47.1667, 98.4274, 45.1976, −84.8263, 111.5884, −80.5106, 91.2663, 48.1103, −112.4206, −57.3296, −1.8639, −112.1757, 104.3520, 12.3912, −18.8735, 105.2388, 43.2676, 24.6232, −3.3290, −8.9063, −105.6909, −59.7953, −5.7724, 3.3887, 18.6702, 5.0219, 0.5422, 14.3978, −16.7167, −104.1545, −57.3927, 100.5543, −98.1572, 59.2683, 103.3772, −107.7848, −41.9981, −27.8649, −57.8695, 12.3373, 61.5242, 111.0124, −30.1124, −53.1460, 16.3327, 19.0595, 29.8188, 87.0224, 84.0159, −46.6717, −9.3582, −40.5922, 0.2572, 27.7654, 22.4121, −110.2285, −61.0410, 42.3490, 8.1843, −39.0598, −21.9064, 37.6060, −111.4066, 87.8032, −19.6168, −77.9707, −83.3925, −10.0607, −25.0223, 110.6321, 61.3093, −4.2647, −0.9471, 26.8395, −21.1663, 111.6304, 59.9060, 18.6659}.

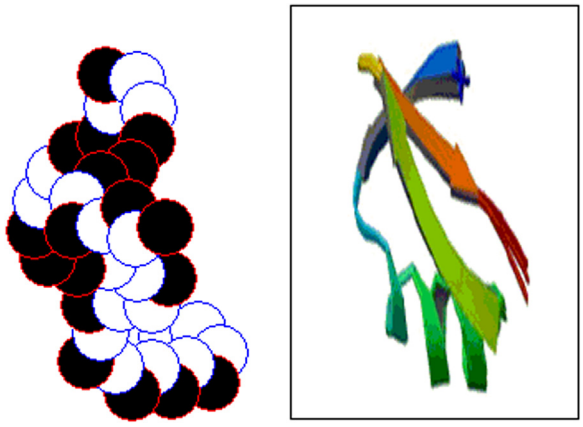

Fig. 16. Comparison between secondary conformation and final functional structure for 1AGT.

results, we can see that IF-ABC obtains lower free-energy values than other approaches. Conformational structures are presented in Figs. 12–15 respectively.

Besides some optimized secondary conformations, their corresponding real structures are also shown in Figs. 16–19, where the amino acid residues are plotted in larger scales to conform to reality. Viewing the real structures of 1AGT and 1AHO, we find that the directional trends and helical characteristics are partly revealed in the optimized secondary conformations. However, for the other two longer sequences, the secondary conformations show fewer visual similarities to their natural structures. A few potential reasons may account for this. It might stem from the complexity of the process to form a functional and real structure, or the deviation of the derived function values from the ground truths. Or, the reason might be that AB off-lattice model does not properly reflect the true characteristics of protein structures, since the AB off-lattice model is simplified by classifying the hydrophilic and hydrophobic residues primarily for effectiveness, but it neglects differences inside either of the two classified groups mentioned above. In fact, hydrophobic residues tend to form a core of minimum surface area that encounters water molecules and should be surrounded by hydrophilic residues.

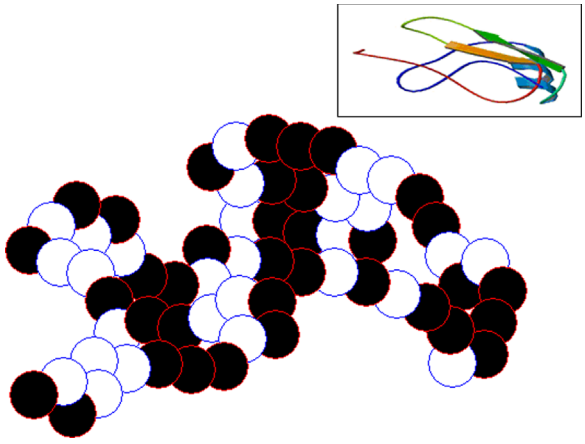

Fig. 17. Comparison between secondary conformation and final functional structure for 1AHO.

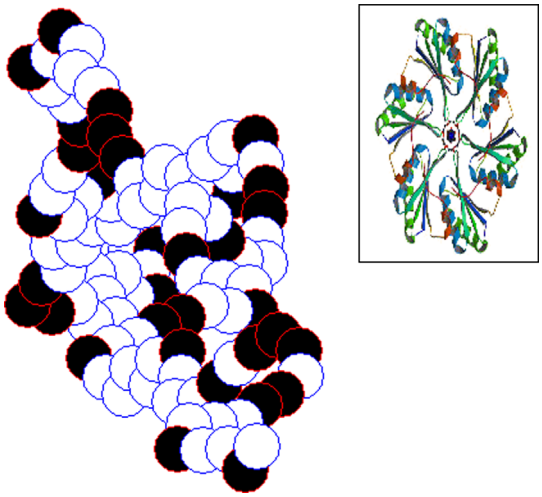

Fig. 18. Comparison between secondary conformation and final functional structure for 2EWH.

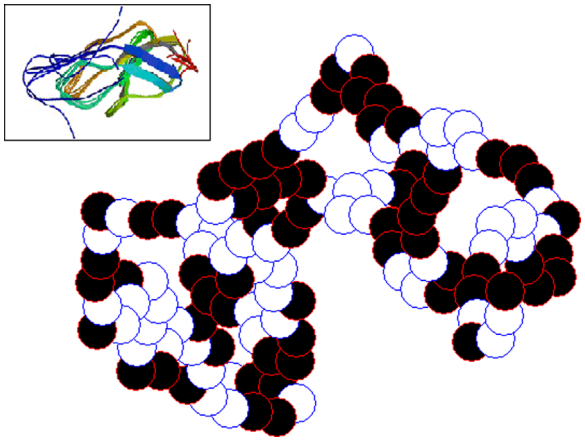

Fig. 19. Comparison between secondary conformation and final functional structure for 2YUX.

6. Conclusion

In this paper, an improved ABC revised by internal feedback strategy is introduced to optimize protein secondary structures in AB off-lattice model. Experimental results confirm that IF-ABC is significantly more effective to improve converging rate than ABC, and is competent for the specific secondary structure prediction problems. It should be noted that our research merely concentrates on sequences with less than 200 amino acids. Optimizations for more natural sequences and investigations on the selection rule of the convergence factor will be our future work.

Acknowledgments

We would like to thank the referees very much for their valuable comments and suggestions. The work was supported in part by the School of Advanced Engineering (SAE) in Beihang University and sponsored by 5th and 6th National College Students' Innovative and Entrepreneurial Training Programs.

Appendix A

In this section, we provide an analytical proof for the convergence of IF-ABC.

For the constrained optimization problem in this paper, it can be considered as a Markov chain model, the state space S is all the possible solutions $\{\mathbf{X}_i\}$, the dimension of S is M. Note that M can be either infinite or finite, which will be discussed later.

There is no harm to assume that elements in $\{\mathbf{X}_i\}$ $(i=1,2,\ldots,M)$ are ranked in descending order by their objective functional values. The probability transition matrix TP of the problem can be written as

$$\begin{pmatrix} tp_{11} & \cdots & tp_{1M} \\ tp_{21} & \cdots & tp_{2M} \\ \vdots & \ddots & \vdots \\ tp_{M1} & \cdots & tp_{MM} \end{pmatrix}, \tag{A.1}$$

where tp_{ij}refers to the probability of transition from the current best solution $\mathbf{X}_i$ to $\mathbf{X}_j$.

Lemma 1. *The greedy selection process in IF-ABC is a stochastic process.*

Proof. In IF-ABC, only the greedy selection process causes transition of optimal solutions. The corresponding transition matrix TP at step t is defined in Eq. (A.1).

It is obvious that each transition probability $tp_{ij} \geq 0$. For each temporarily fixed $\mathbf{X}_i$ at step t, the possibilities to $\mathbf{X}_j, j = 1, 2, \ldots, M$ in the next step turn out to be an inevitable event, i.e.,

$$\sum_{j=1}^{M} tp_{ij} = 1 \quad (i = 1, 2, \ldots, M).$$

Therefore, TP is a stochastic matrix, and the transition process of optimal solutions in IF-ABC is a stochastic process. □

Lemma 2. *The transition process of optimal solutions in IF-ABC is a finite homogeneous Markov process.*

Proof. It is clear that the dimension of the state space S is finite, so the Markov chain is finite. By employing Eqs. (4) and (5) in the process, we observe that the selection process in the next step only depends on the current state, and is independent from the step t. It follows that the process is homogeneous. □

Lemma 3. *Let $TP(t)$ be the transition matrix at step t, $TP(k)TP(k+1) = TP(k+2)$.*

Proof. Denote entries in $TP(t+k)$ by tp_{ij}^{k}, $k \geq 0$ $(i, j = 1, 2, \ldots, M)$. By Lemma 2, any pair of probabilities from different steps is independent.

The entry located in the ith row and jth column of $TP(k)TP(k+1)$ is

$$\sum_{m=1}^{M} tp_{im}^{k} tp_{mj}^{k+1}.$$

Recall that $tp_{im}^{k} tp_{mj}^{k+1}$ denotes the probability of the two-step transition process from $\mathbf{X}_i$ to $\mathbf{X}_m$ and then to $\mathbf{X}_j$. All such processes sum up to be an inevitable event, which can be considered as a one-step process from $\mathbf{X}_i$ directly to $\mathbf{X}_j$. Therefore, the transition probability equals to tp_{ij}^{k+2}, i.e., $TP(k)TP(k+1) = TP(k+2)$. □

The conclusion can be extended to

$$TP(k) = \prod_{i=1}^{k} TP(i).$$

Lemma 4. (Rudolph, 1994). *Let W be a reducible stochastic matrix, i.e., it can be brought into the form (with square matrices C and T)*

$$\begin{pmatrix} C & \mathbf{0} \\ R & T \end{pmatrix}$$

by applying the same permutations to rows and columns. If C is a primitive stochastic matrix and $R, T \neq \mathbf{0}$, then

$$\lim_{k\to\infty} W^k = \lim_{k\to\infty} \begin{pmatrix} C^k & \mathbf{0} \\ \sum_{i=0}^{k-1} T^i R C^{k-i} & T^k \end{pmatrix} = \begin{pmatrix} C^{\infty} & \mathbf{0} \\ R_{\infty} & \mathbf{0} \end{pmatrix}$$

is a stable stochastic matrix.

For engineering problems running on computers, continuous domains are discretized due to finite word lengths in computer. Therefore, the corresponding M is finite.

Theorem. *The process of IF-ABC converges to the global optimum in a discrete domain.*

Proof. Recall that the greedy selection strategy implies that $tp_{mn} = 0$ for any $m > n$. Hence, at step t, the transition matrix of optimal solutions $TP(t)$ is

$$TP(t) = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ tp_{21} & tp_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ tp_{M1} & tp_{M2} & \cdots & tp_{MM} \end{pmatrix}.$$

This matrix is reducible, since it can be expressed as a lower triangle matrix

$$\begin{pmatrix} 1 & \mathbf{0} \\ R_{(M-1)\times 1} & T_{(M-1)\times(M-1)} \end{pmatrix}.$$

Besides, it follows from Lemma 2 that the process is a finite homogeneous Markov process regardless of step t. Thus, by applying Lemma 3, we get

$$\lim_{k\to\infty} TP(k) = \lim_{k\to\infty} \prod_{i=1}^{k-1} TP(i).$$

Moreover, since $TP(t)$ is stochastic, then Lemma 4 yields that

$$\lim_{k\to\infty} TP(k) = \begin{pmatrix} 1 & \mathbf{0} \\ R_{\infty} & \mathbf{0} \end{pmatrix}$$

is also a stochastic matrix. It follows that each row of the above matrix sums up to 1.Hence,

$$R_{\infty} = \begin{bmatrix} tp_{21} \\ \vdots \\ tp_{N1} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Consequently, each possible solution as a current best solution will approach towards the global optimum with probability 1 as the step $t \to \infty$. In other words, when t is large enough, it is certain for the searching process to help to find the global optimum at the $(t+1)$th step. □

References

Anfinsen, C.B., 1973. Principles that govern the folding of protein chains. Science 181 (96), 223–230.

Berger, B., Leighton, T., 1998. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. J. Comput. Biol. 5 (1), 27–40.

Chiu, T.K, Kubelka, J., Herbst-Irmer, R., Eaton, W.A., Hofrichter, J., Davies, D.R., 2005. High-resolution x-ray crystal structures of the villin headpiece subdomain, an ultrafast folding protein. Proc. Natl. Acad. Sci. USA 102 (21), 7517–7522.

Clark, A.G., 2007. Evolution of genes and genomes on the Drosophila phylogeny. Nature 450 (7167), 203–218.

Cole, C., Barber, J.D., Barton, G.J., 2008. The Jpred 3 secondary structure prediction server. Nucleic Acids Res. 36, W197–W201.

Cutello, V., Nicosia, G., Pavone, M., Timmis, J., 2007. An immune algorithm for protein structure prediction on lattice models. IEEE Trans. Evol. Comput. 11 (1), 101–117.

Gao, W., Liu, S., 2011. Improved artificial bee colony algorithm for global optimization. Inform. Process. Lett. 111 (17), 871–882.

Gozuacik, D., Adi, K., 2004. Autophagy as a cell death and tumor suppressor mechanism. Oncogene 16 (16), 2891–2906.

Holland, J.H., 1992. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. University of Michigan Press, Oxford, England.

Hsu, H-P, Mehra, V., Grassberger, P., 2003. Structure optimization in an off-lattice protein model. Phys. Rev. E 68 (3), 037703.

Joo, K., Kim, I., Kim, S.Y., Lee, J., Lee, J., Lee, S.J., 2004. Prediction of the secondary structure of proteins using PREDICT, a nearest neighbor method on pattern space. J. Kor. Phys. Soc. 45, 1441–1449.

Kang, F., Li, J., Li, H., 2013. Artificial bee colony algorithm and pattern search hybridized for global optimization. Appl. Soft Comput. 13 (4), 1781–1791.

Karaboga, D., 2005. An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-TR06. Erciyes University, Engineering Faculty, Computer Engineering Department.

Kennedy, James, Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, 1995, Piscataway, NJ, pp. 1942–1948.

Kim, S-Y., Lee, S.B., Lee, J., 2005. Structure optimization by conformational space annealing in an off-lattice protein model. Phys. Rev. E 72 (1), 011916.

Li, B., Li, Y., 2012. BE-ABC: hybrid artificial bee colony algorithm with balancing evolution strategy. In: 2012 Third International Conference on Intelligent Control and Information Processing, pp. 217–222.
Li, C., Ding, Y., Xu, W., 2010. Multiple-layer quantum-behaved particle swarm optimization and toy model for protein structure prediction. In: 2010 Ninth International Symposium on Distributed Computing and Applications to Business Engineering and Science (DCABES), pp. 92–96.
Li, Y., Liu, H., Rata, I., Jakobsson, E., 2013. Building a knowledge-based statistical potential by capturing high-order inter-residue interactions and its applications in protein secondary structure assessment. J. Chem. Inform. Model. 53 (2), 500–508.
Lin, X., Zhu, H., 2008. Structure optimization by an improved tabu search in the AB off-lattice protein model. In: Intelligent. ICINIS'08 First International Conference on Networks and Intelligent Systems, pp. 123–126.
Liu, J., Wang, L., He, L., Shi, F., 2005. Analysis of toy model for protein folding based on particle swarm optimization algorithm. In: ICNC'05 Proceedings of the First International Conference on Advances in Natural Computation, vol. 3, pp. 636–645.
Malekpour, S.A., Naghizadeh, S., Pezeshk, H., 2009. Protein secondary structure prediction using three neural networks and a segmental semi Markov model. Math. Biosci. 217 (2), 145–150.
Manuel, M., Elias, E., 2013. Design of frequency response masking FIR filter in the Canonic Signed Digit space using modified Artificial Bee Colony algorithm. Eng. Appl. Artif. Intell. 26 (1), 660–668.
McGuffin, L.J., Bryson, K., Jones, D.T., 2000. The PSIPRED protein structure prediction server. Bioinformatics 16 (4), 404–405.
Montgomerie, S., Sundararaj, S., Gallin, W.J., Wishart, D.S., 2006. Improving the accuracy of protein secondary structure prediction using structural alignment. BMC Bioinform. 7, 301.
Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C., 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. 247 (4), 536–540.
Omkar, S.N., Senthilnath, J., Khandelwal, R., Narayana Naik, G., Gopalakrishnan, S., 2011. Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures. Appl. Soft Comput. 11 (1), 489–499.
Pollastri, G., Martin, A.J., Mooney, C., Vullo, A., 2007. Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. BMC Bioinform. 8, 201.
Przybylski, D., Rost, B., 2002. Alignments grow, secondary structure prediction improves. Proteins: Struct.Funct. Bioinform. 46 (2), 197–205.
Pu, C. D., 2011. Improvement of PSO and its Application in Protein Folding Prediction. Wuhan University of Science and Technology, Wuhan (in Chinese).
Rudolph, G., 1994. Convergence analysis of canonical genetic algorithms. IEEE Trans. Neural Networks 5 (1), 96–101.
Shmygelska, A., Hoos, H.H., 2005. An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. BMC Bioinform. 6 (1), 30.
Stillinger, F.H., Head-Gordon, T., Hirshfeld, C.L., 1993. Toy model for protein folding. Phys. Rev. E 48 (2), 1469–1477.
Storn, R., Price, K., 1995. Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. Technical Report TR-95-012. Berkeley, CA.
Wang, T., Zhang, X., 2009. 3D Protein structure prediction with genetic tabu search algorithm in Off-Lattice AB model. In: KAM'09 s International Symposium on Knowledge Acquisition and Modeling, vol. 1, pp. 43–46.
Xiang, W., An, M., 2012. An efficient and robust artificial bee colony algorithm for numerical optimization. Comput. Oper. Res. 40 (5), 1256–1265.
Zhang, X., Lin, X., 2006. Effective protein folding prediction based on genetic-annealing algorithm in Toy model. In: 2006 Workshop on Intelligent Computing & Bioinformatics of CAS.
Zhou, X.B., Chen, C., Li, Z.C., Zou, X.Y., 2008. Improved prediction of subcellular location for apoptosis proteins by the dual-layer support vector machine. Amino Acids 35 (2), 383–388.
Zhou, Y., Han, P., 2010. Analysis of protein folding using a novel hybrid evolutionary algorithm. China J. Bioinform. 8 (1), 73–75. (in Chinese).