International Journal of Innovative Computing, Information and Control
Volume **8**, Number **3(B)**, March **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **2049–2064**

USING AN EFFICIENT ARTIFICIAL BEE COLONY ALGORITHM FOR PROTEIN STRUCTURE PREDICTION ON LATTICE MODELS

Cheng-Jian Lin[1] and Shih-Chieh Su[2]

[1]Department of Computer Science and Information Engineering
National Chin-Yi University of Technology
No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan
cjlin@ncut.edu.tw

[2]Department of Computer Science and Information Engineering
National Chung Cheng University
No. 168, University Rd., Chiayi County 62102, Taiwan
ssc95p@cs.ccu.edu.tw

Received December 2010; revised April 2011

Abstract. *The well-known artificial bee colony (ABC) algorithm is one of the most recently introduced swarm-based algorithms. The ABC system combines both local and global search methods in an attempt to balance exploration and exploitation processes, and hopefully, it can be successfully applied to solve real-world problems. In the past, the Science Magazine named the protein folding problem (PFP) as one of the 125 biggest unsolved problems in science. The PFP addresses the question of how the amino acid sequence (AAS) of a specific protein dictates its structure. In the study, we present a modified ABC (MABC) algorithm for the PFP with both 2D and 3D HP models. We demonstrate that our algorithm can be applied successfully to the protein folding problem based on the hydrophobic-polar lattice model. The simulation results show that the modified artificial bee colony algorithm can successfully be applied to the protein folding problem.*

Keywords: Protein folding, Protein structure prediction, HP-model, Swarm intelligence, Artificial bee colony algorithm

1. **Introduction.** Artificial bee colony (ABC) algorithm is one of the most recently introduced swarm-based algorithms [1]. In ABC system, artificial bees fly around in a multidimensional searching space and some (employed and onlooker bees) choose food sources depending on their own experience and that of their nest mates, and adjust their positions. That means ABC is an optimal tool providing a population-based searching procedure in which individuals called food-position are modified by the artificial bees with time and the bees' targets are to find the food sources with high nectar amount, the place with the highest nectar. (i.e., the best solution.) It is as simple as particle swarm optimization (PSO) [2] and differential evolution (DE) [3], but only uses two common control parameters namely colony size and maximum cycle number.

ABC system combines local and global search methods attempting to balance exploration and exploitation process. Hopefully, it can be applied to solve the real world-problems. Karaboga and Basturk [4] compare the performance of ABC algorithm with that of differential evolution (DE), particle swarm optimization (PSO) and evolutionary algorithm (EA) for multi-dimensional numeric problems. Simulation results show that ABC algorithm performs better than the mentioned algorithms and can be effectively employed to solve the multimodal engineering problems with high dimensionality.

Karaboga and Akay [5] use ABC for optimizing a large set of numerical test functions and the results produced by ABC algorithm are compared with the results obtained by genetic algorithm, particle swarm optimization algorithm, differential evolution algorithm and evolution strategies. The results show that the performance of the ABC similar is better than or to the other population-based algorithms with the advantage of employing fewer control parameters. Karaboga and Ozturk [6] are applying ABC algorithm to clustering analysis, as 13 typical swarm-off data for study. The results of the experiment show that the artificial bee colony algorithm can successfully be applied to cluster for the purpose of the classification.

Karaboga and Basturk have extended the ABC algorithm for constrained optimization problems in [7] and applied ABC for training neural networks [8,9]. Moreover, the ABC algorithm was used for designing IIR filters in [10] and for the leaf-constrained minimum spanning tree problem in [11], etc., all of which indicate ABC algorithm certainly can be used to solve some real application problems.

In 2005, Science named the protein folding problem as one of the 125 biggest unsolved problems in science [12]. The protein folding problem is the question of how the amino acid sequence of a protein dictates its structure, because the amino acid sequence of a protein determines its structure which determines its mechanism of action. This key paradigm in biochemistry accounts for nearly one fourth Nobel Prizes in Chemistry since 1956 [13].

Currently, protein structures are primarily determined by techniques such as MRI (magnetic resonance imaging) [16,17] and X-ray crystallography [14,15], which is expensive in terms of equipment, computation and time. Additionally, these techniques require isolation, purification and crystallization of the target protein. The difficulty in solving protein structure prediction problems stems from two major sources: (1) finding good measures for the quality of candidate structures and (2) given such measures, determining optimal or close-to-optimal structures for a given amino-acid sequence [18]. Therefore, computational approaches to protein structure prediction are very attractive.

The HP model (Hydrophobic-Polar [19]) is a simplified model which has become very popular. The HP model is one of the most widely used models. However, the HP model has been shown to be NP-complete on the square lattice [20] and cubic lattice [21], so many approximation and heuristic search algorithms have been developed for a variety of lattice models [18,22-29]. Unger and Moult [18] described a genetic algorithm (GA) application that used heuristic-based crossover and mutation operators to solve the HP model. Jiang et al. [24] presented a hybrid algorithm combining genetic algorithms and tabu search (GTS) which is the first work of hybridizing TS and GA for the protein folding based on the HP model. Shmygelska, Hernandez and Hoos [25] proposed an ACO algorithm which is the first application of ACO to this problem and an improved ACO [26] for the 2D HP protein folding problem. This is also the first method applying the ACO algorithm to 2-dimensional (2D) and 3-dimensional (3D) HP model [27]. Recently, Cutello et al. [28] presented an immune algorithm (IA) for the protein folding based on the HP model, and also Santana, Larrañaga and Lozano [29] presented an estimation of distribution algorithms (EDAs) or the protein folding based on the HP model.

In the study, we present a modified artificial bee colony (MABC) algorithm for the 2D and 3D HP model, and make a comparison with the above-mentioned methods. In Section 2, we briefly introduce the HP model. In Section 3, the foraging behaviour of real bees and then ABC algorithm simulating this behaviour are described. Finally, in Section 4 and Section 5, the simulation results obtained are presented and discussed respectively.

2. **Preliminaries.** In this section, we briefly present the HP protein folding problem and its free energy calculation.

2.1. **The HP protein folding problem.** The principle of HP model relies on the hydrophobic interaction as the main driving force in protein folding. One of the main driving forces of folding in globular proteins is the hydrophobic interaction, which tends to pack hydrophobic amino acids in the center of the protein. Hydrophobicity is one of the key factors that determine how the chain of amino acids will fold up into an active protein.

In the HP model, each amino acid is represented as a bead, and connecting bonds are represented as lines. In this approach, the protein is composed of a specific sequence of only two types of beads, H (bead-Hydrophobic/non-polar) or P (bead-hydrophilic/Polar); that is; the 20 amino acids can be divided into two classes: H and P.

As Hockenmaier, Joshi and Dill [30] thought, there are three advantages of HP model: (1) the full conformational space of a protein molecule can be completely enumerated, so the native state can be known with no assumption or approximation, (2) the model has the same NP-complete search problem as in real proteins, where the size of the search space grows exponentially with chain length, and yet, for many sequences, there is only a single native structure, and (3) despite its simplicity, it captures the physics of hydrophobic inter-actions, steric excluded volume, and chain conformational freedom that are key components of real protein folding. Thus, the HP model allows much more extensive and unambiguous testing of search strategies than other models.

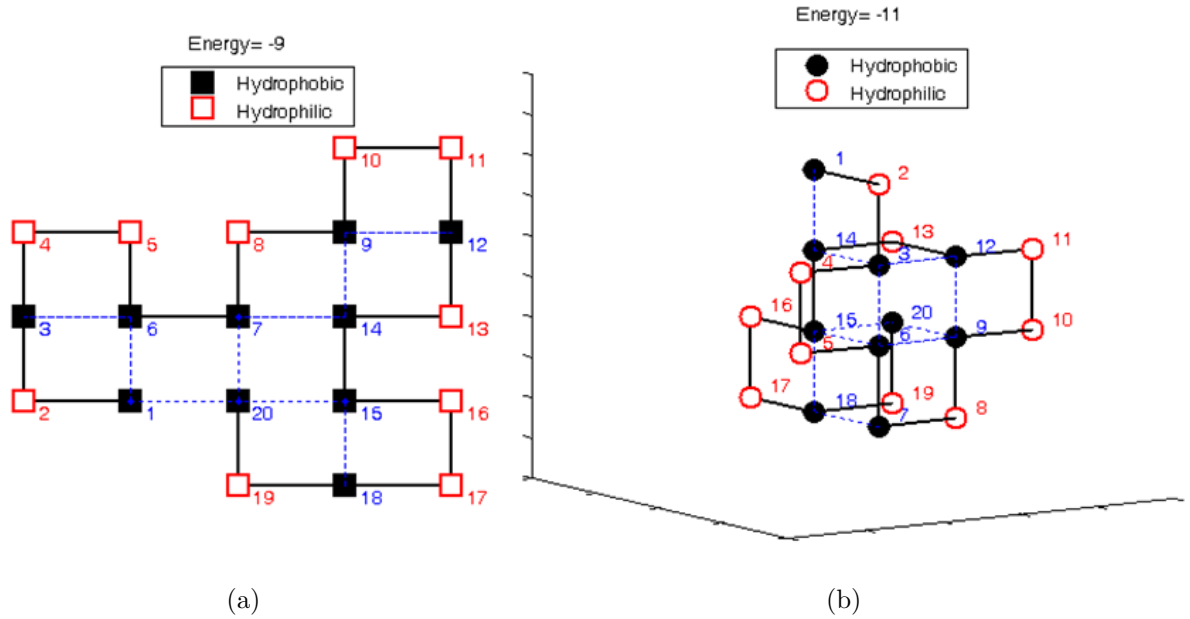
An instance is shown in Figure 1 for the 2D and 3D HP lattice model from our benchmarked [24], respectively. The black squares denote the hydrophobic amino acid and the white squares denote the hydrophilic. The dotted line denotes the H-H contacts (free energy) in the conformation, which are assigned an energy value of -1. The free energy is minimum value; the number of H-H contact is the maximum. In two-dimensional case, Figure 1(a) shows a protein structure with 9 H-H contacts (energy $= -9$). In three-dimensional case, Figure 1(b) shows a protein structure with 11 H-H contacts (energy $= -11$). Since the native state of a protein generally corresponds to the lowest free energy state for the protein, the optimal conformation in the HP model is the one that has the maximum number of H-H contacts which gives the lowest energy value.

2.2. **Calculating the free energy.** One of the most popular lattice models, the HP model, features just two bead types: H (hydrophobic or non-polar) and P (hydrophilic or polar). An instance is shown in Figure 1 for the 2D and 3D HP lattice model. The black beads denote the hydrophobic amino acid and white beads denote the hydrophilic. The dotted line denotes the H-H contacts (free energy) in the conformation, which are assigned an energy value of -1. The free energy is minimum; the number of H-H contacts is the maximum. Figure 1(a) shows a protein structure with 9 H-H contacts (energy $= -9$). Figure 1(b) shows a protein structure with 11 H-H contacts (energy $= -11$). Since the native state of a protein generally corresponds to the lowest free energy state for the protein, the optimal conformation in the HP model is the one that has the maximum number of H-H contacts which gives the lowest energy value.

The free energy for the protein can be calculated by the following formulae [31],

$$\epsilon_{ij} = \begin{cases} -1.0 & \text{the pair of H and H residues} \\ 0.0 & \text{others} \end{cases} \tag{1}$$

$$E = \sum_{i,j} \triangle r_{ij} \epsilon_{ij} \tag{2}$$



(a) (b)

FIGURE 1. An optimal conformation for the sequence "(HP)2PH(HP)2 (PH)2HP(PH)2"; (a) the 2D HP lattice model and (b) the 3D HP lattice model

where the parameter

$$\triangle r_{ij} = \begin{cases} 1 & S_i \text{ and } S_j \text{ are adjacent but not connected amino acids} \\ 0 & \text{others} \end{cases} \tag{3}$$

Hence, the protein folding problem can be transformed into an optimization problem, i.e., to calculate the minimal free energy of the protein folding conformation. As a result, the following problem can be formally defined as: "given an HP sequence $s = s_1, s_2, \ldots, s_n$, find an energy-minimizing conformation of s; that is, find $c^* \in C(s)$ such that $E(c^*) = \min\{E(c)|c \in C\}$, where $C(s)$ is the set of all valid conformations for s [32]."

3. **Methods.**

3.1. **Behaviour of real bees.** The artificial bee colony (ABC) algorithm is a new swarm intelligence based optimizer proposed by Karaboga [1] for multivariable and multi-modal continuous function optimization. Inspired by the intelligent foraging behavior of honeybee colony [34-36]. Tereshko consists of three essential components: food sources, employed foragers and unemployed foragers, and defines two leading modes of the behaviour: recruitment to a nectar source and abandonment of a source.

Food Sources: The value of a food source to an insect depends on many factors including its proximity to the nest, richness or concentration of energy, and the ease of extracting this energy. The key point is to describe the "profitability" of a food source with a single quantity and to see how insects react to food sources with different values of this quantity, if they always are able to select the "best" food source in a changing environment.

Employed Foragers: Employed foragers are associated with a particular food source which they are currently exploiting or are "employed" at. They carry with them information about this particular source, its distance and direction from the nest, and the profitability of the source. Employed foragers will share this information with a certain probability. The greater the profitability of a food source, the higher the probability the

honeybee will do a waggle dance and share her information with her nest mates. Note, however, that employed foragers are only locally informed – they know only of the food source they are currently exploiting and continue frequenting this food source until it is depleted, at which point they become unemployed foragers.

Unemployed Foragers: Unemployed foragers are looking for a food source to exploit. There are two types of unemployed foragers, scouts, who search the environment surrounding the nest (up to a 14 km radius) in search of new food sources, and onlookers who wait in the nest and find a food source through the information shared by employed foragers. The percentage of unemployed foragers who are scouts varies from 5% to as much as 30% depending on the influx of information into the nest. The mean number of scouts averaged over conditions is about 10%.

Karaboga [1] described Tereshko model as Behaviour of real bees shown in Figure 2 to explain the connection of the aforementioned models.

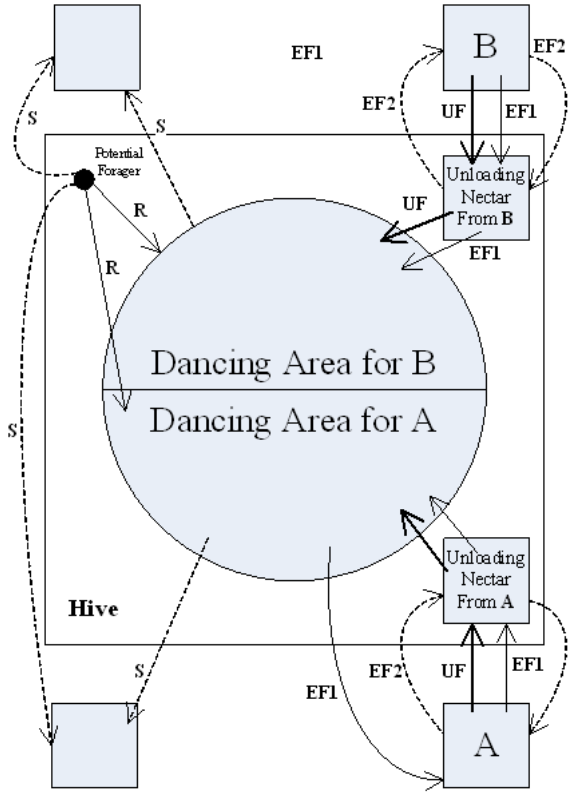


FIGURE 2. Behaviour of honeybee foraging for nectar

Assume that there are two discovered food sources: A and B.

At the very beginning, a potential forager will start as unemployed forager. That bee will have no knowledge about the food sources around the nest.

There are two possible options for such a bee:

(1) It can be a scout and starts searching around the nest spontaneously for food due to some internal motivation or possible external clue ('S' in Figure 2).
(2) It can be a recruit after watching the waggle dances and starts searching for a food source ('R' in Figure 2).

After finding the food source, the bee utilizes its own capability to memorize the location and then immediately starts exploiting it. Hence, the bee will become an "employed forager". The foraging bee takes a load of nectar from the source and returns to the hive, unloading the nectar into storage.

After unloading the food, the bee has the following options:

(1) It might become an uncommitted follower after abandoning the food source (UF).
(2) It might dance and then recruit nest mates before returning to the same food source (EF1).
(3) It might continue to forage at the food source without recruiting other bees (EF2).

It is important to note that not all bees start foraging simultaneously. The experiments confirmed that new bees begin foraging at a rate proportional to the difference between the eventual total number of bees and the number presently foraging.

3.2. **Artificial bee colony (ABC) algorithm.** In ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts.

First half of the colony consists of the employed artificial bees and the second half includes the onlookers. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources. The employed bee of an abandoned food source becomes a scout. The search carried out by the artificial bees can be summarized as follows:

(1) Employed bees determine a food source within the neighbourhood of the food source in their memory.
(2) Employed bees share their information with onlookers within the hive and then the onlookers select one of the food sources.
(3) Onlookers select a food source located within their proximity.
(4) An employed bee of which the source has been abandoned becomes a scout and starts to search a new food source randomly.

The main steps of the algorithm are given below:

1. INITIALIZE
2. **REPEAT**
3. Move the employed bees onto their food sources and determine their nectar amounts.
4. Move the onlookers onto the food sources and determine their nectar amounts.
5. Move the scouts for searching new food sources.
6. Memorize the best food source found so far.
7. **UNTIL** (requirements are met)

Each cycle of the search consists of three steps: moving the employed and onlooker bees onto the food sources and calculating their nectar amounts and determining the scout bees and then moving them randomly onto the possible food sources. A food source represents a possible solution to the problem to be optimized. The nectar amount of a food source corresponds to the quality of the solution represented by that food source.

3.3. **The modified artificial bee colony algorithm.** In order to effectively apply Artificial Bee Colony algorithm into the protein folding problem, we proposed the Modified Artificial Bee Colony (MABC) algorithm by the following descriptions.

3.3.1. *Initialization step.* Within the initialization step, the number of colony is equal to the sum of employed and onlooker bees, and the number of food sources is the half of the colony size. In this work, the term "population" indicates food sources. When the system starts operating, an initial population is generated and the fitness value of each initial population is then evaluated, while the detailed processes are explained in the following paragraphs.

Initial population: If the input amino-acid sequence is of length n, then each individual in the population is a string of length $(n-1)$ over the symbols $= \{U, L, R, D\}$

and that denotes a valid conformation in the 2D square lattice. The symbols U, L, R and D are used to denote the fold directions *up*, *left*, *right* and *down* in the encoding scheme, respectively. An initial population is generated randomly and initializes an $(n-1)$-dimensional space within a fixed range. As depicted in Figure 3, it adopted related schemes for representing internal movements in absolute directions.

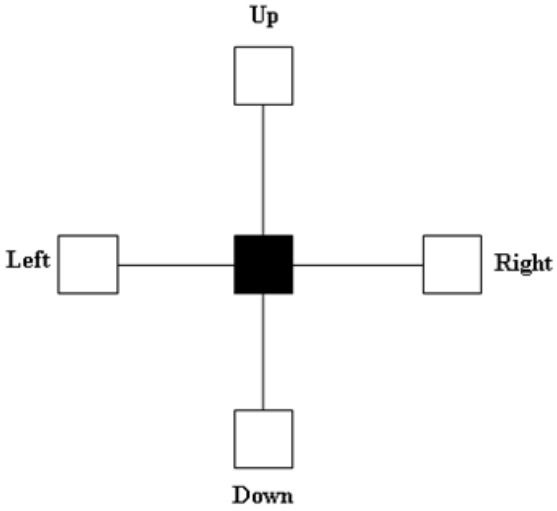


FIGURE 3. Representation of internal coordinates (the black cube is the current location)

In the 3D case, the symbols R, L, F, B, U and D are used to denote the fold directions *right*, *left*, *forward*, *backward*, *up* and *down* in the encoding scheme, respectively. An initial population is generated randomly and initializes an $(n-1)$-dimensional space within a fixed range. As illustrated in Figure 4, the internal movements are represented in absolute directions. An initial population is generated randomly and initializes an $(n-1)$-dimensional space within a fixed range.

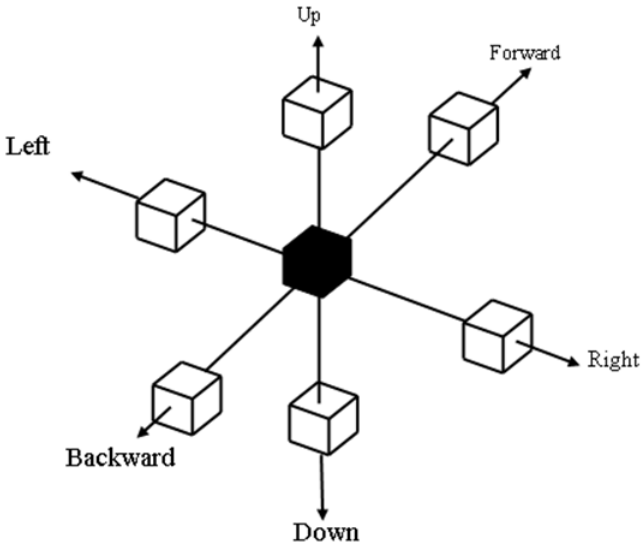


FIGURE 4. The schemes were represented by internal coordinates (the black cube represents the current location)

Evaluating: The evaluating step evaluates each chromosome in a population. Since the goal of the MABC method is to minimize the fitness value, the lower a fitness value, the better the fitness. The fitness function is defined by Equations (1)-(3).

3.3.2. *Repeat step.* Repeat the following steps.

3.3.3. *Employed step.* Move the employed bees onto their food sources and determine their nectar amounts.

For every food source, there is a pair of employed bees responsible for nectar collecting and for sharing their information with onlookers within the hive and then the onlookers select one of the food sources.

3.3.4. *Onlookers step.* Move the onlookers onto the food sources and determine their nectar amounts.

The onlookers will search for food resources nearby according to the information provided by the employed bees. Onlookers are placed on the foods by using "roulette wheel selection" method [37], as defined as in Equation (4),

$$P_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i}. \tag{4}$$

Onlookers can perform an intensive search for a new and better solution. This is similar to the mutation operation. In other words, the onlookers play the role of performing local search. The local search is different from the mutation operation in term of the rules. A local search has its system rules and is able to effectively find a local solution. The two components of local searches are described in the following.

(1) Opposite Motion

As shown in Figure 5, the motion of a local structure is illustrated in the 2D HP model. We can change the local structure between two randomly determined sequence positions. The second residue to the fifth residue directions are *right*, *down*, *right* into *left*, *up*, *left*, respectively. The inversion method can advance in the opposite direction.

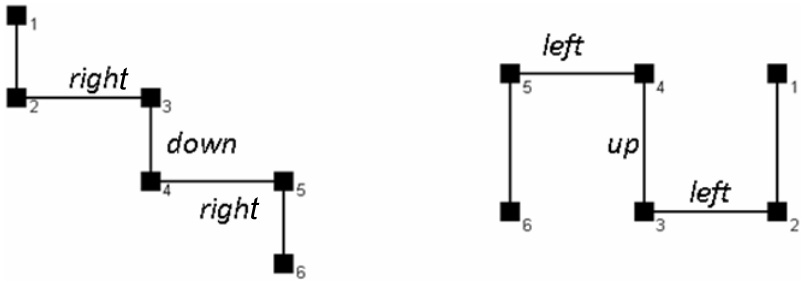


FIGURE 5. Example of the opposite motion: the second to fifth residues represent opposite position

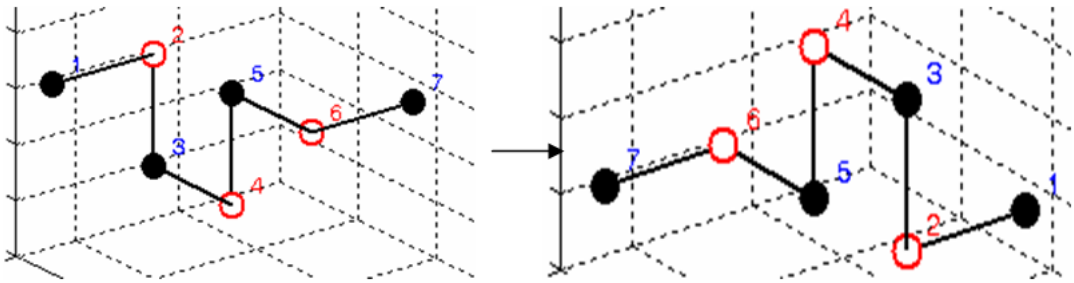


FIGURE 6. All the residues are moved to the opposite direction

As depicted in Figure 6, the motion of a local structure is in the 3D HP model. Similar to the previous example, we can change the local structure between two randomly determined sequence positions. All the residue directions are *right*, *down*, *backward*, *up*, *backward*, *right* to *left*, *up*, *forward*, *down*, *forward* and *left*. The inversion method can be preceded in the opposite direction, which is the direction of repulsion.

(2) Rotation Motion

The rotation motion is divided into rotation clockwise (CW) and counter-clockwise (CCW). As presented in Figure 7. The second to the fifth residue directions are *right*, *down*, *right* into *down*, *left* and *down*, respectively.

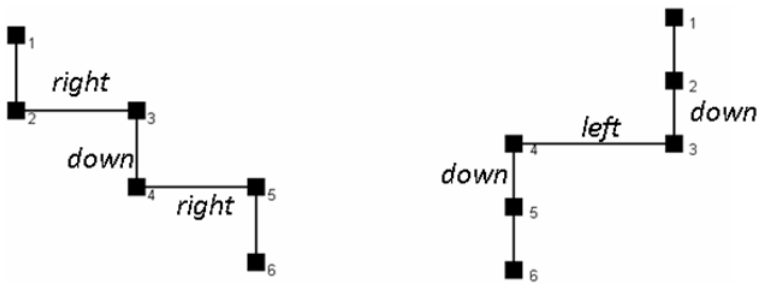


FIGURE 7. Rotation motion: the second to fifth residues will move to clockwise position and form the structure as the right one

In a 3D case, the structure can rotate into CW or CCW relative to the local structure, as demonstrated in Figures 8 and 9. Figures 8 and 9 represent CW and CCW rotations, respectively. The illustrations indicate that the left-to-right transformation from the top is respectively the fixed x-, the fixed y- and the fixed z-axis.

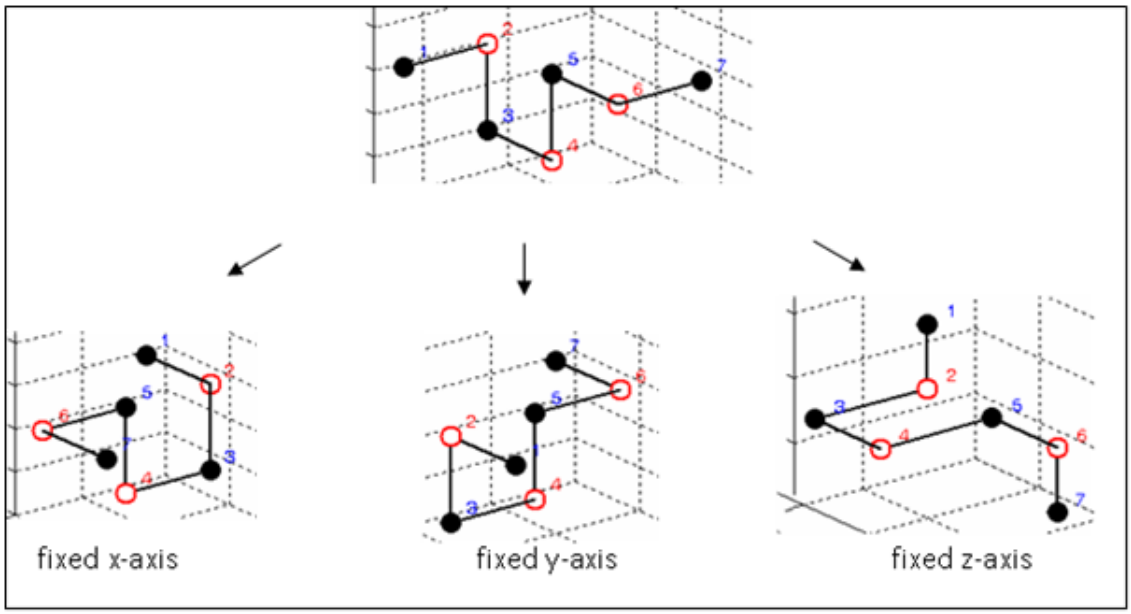


FIGURE 8. The clockwise rotation motion

Therefore, we can make the summarization as in Tables 1 and 2, which list the relationship between the original and transformed directions. Table 1 denotes the residue folding direction with local search of the 2D HP model, while Table 2 is for the 3D HP model. We then choose the best direction in a local search by above-mentioned three approaches. As the result, the new folding direction is superior to the original direction.

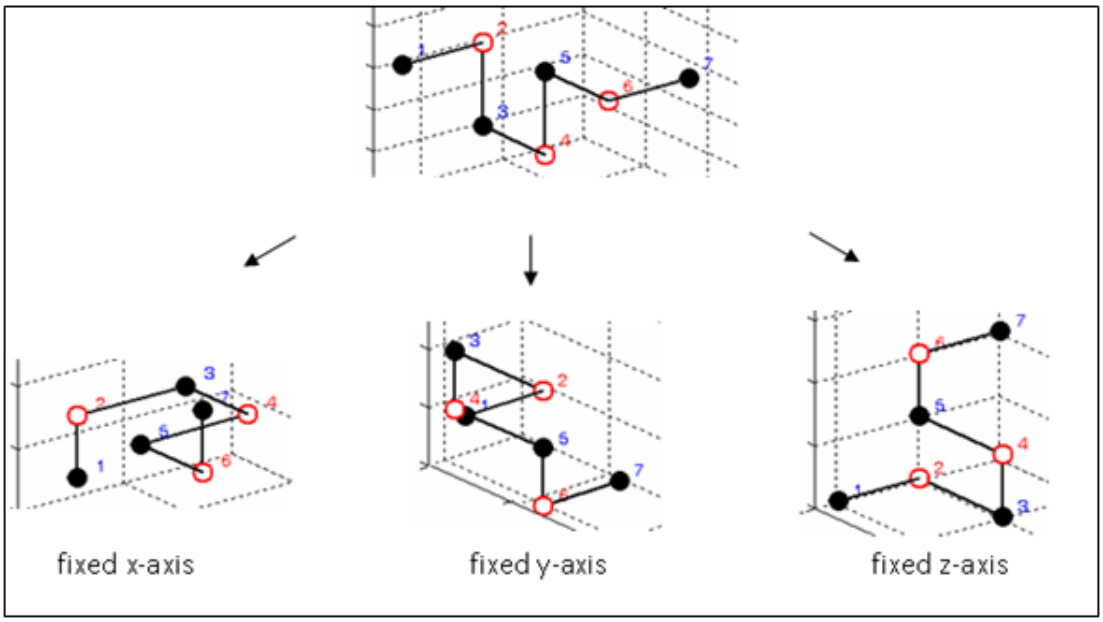


FIGURE 9. The counterclockwise rotation motion

If the new folding direction is not better than the original one, the original direction will not be changed.

TABLE 1. The residue folding direction with local search: the 2D case

Original direction	Opposite direction	CW direction	CCW direction
Right (*R*)	*L*	*D*	*U*
Left (*L*)	*R*	*U*	*D*
Up (*U*)	*D*	*R*	*L*
Down (*D*)	*U*	*L*	*R*

TABLE 2. The residue folding direction with local search: the 3D case

Direction	Opposite	The z-axis fixed		The y-axis fixed		The x-axis fixed	
		CW	CCW	CW	CCW	CW	CCW
Right (*R*)	*L*	*B*	*F*	*D*	*U*	*R*	*R*
Left (*L*)	*R*	*F*	*B*	*U*	*D*	*L*	*L*
Forward (*F*)	*B*	*R*	*L*	*F*	*F*	*U*	*D*
Backward (*B*)	*F*	*L*	*R*	*B*	*B*	*D*	*U*
Up (*U*)	*D*	*U*	*U*	*R*	*L*	*B*	*F*
Down (*D*)	*U*	*D*	*D*	*L*	*R*	*F*	*B*

3.3.5. *Scouts step: Move the scouts for searching new food sources.* The scouts search for food sources in a random way and they do not use the information from the employed bees. When the number of the employed bees exceeds certain threshold or the employed bees decide to give up the food sources, they will become scouts and search the food in a random way at a average cost of a low in food source quality.

3.3.6. *Memorize the best food source found so far.* In this step, the local and global bests are updated. If the fitness value of a particle is higher than that of the local best, then the local best will be replaced with the particle; and if the local best is better than the current global best, than the global best is replaced with the local best in the swarm.

3.3.7. *Termination condition.* The proposed algorithm is run for a predefined maximum of iterations. The best solution of the population is then returned.

4. **Simulation Results.** Our MABC algorithm is compared with the traditional genetic algorithm (GA) [18], the ant colony optimization (ACO) algorithm [25], Monte Carlo (MC) [38], tabu search with the genetic algorithm (GTS) [24], immune algorithm (IA) [28] and estimation of distribution algorithms (EDAs) [29]. In Table 3, the 8 chosen HP instances are standard benchmarks used to test the searching ability of the algorithms. The free energy is the optimal or best-known energy value. $Hi, Pi\,e(\ldots)i$ indicates i repetitions of the relative symbol or subsequence. Sequences 1 through 8 were introduced in [24]. These sequences have been used as the benchmark for the 2D HP model.

Table 3. The 2D HP benchmarks

Seq.	Length	Protein Sequence	Energy (2D)
1	20	$(HP)^2PH(HP)^2(PH)^2HP(PH)^2$	-9
2	24	$H^2P^2(HP^2)^6H^2$	-9
3	25	$P^2HP^2(H^2P^4)^3H^2$	-8
4	36	$P(P^2H^2)^2P^5H^5(H^2P^2)^2P^2H(HP^2)^2$	-14
5	48	$P^2H(P^2H^2)^2P^5H^{10}P^6(H^2P^2)^2HP^2H^5$	-23
6	50	$H^2(PH)^3PH^4PH(P^3H)^2P^4(HP^3)^2HPH^4(PH)^3PH^2$	-21
7	60	$P(PH^3)^2H^5P^3H^{10}PHP^3H^{12}P^4H^6PH^2PHP$	-36
8	64	$H^{12}(PH)^2((P^2H^2)^2P^2H)^3(PH)^2H^{11}$	-42

We give the structure obtained by our algorithm as follows. Fifty independent runs of the algorithms were performed. For sequences 1 through 3, a population size of 100 was used. For sequences 4 through 6, a population size of 200 was used. Sequences 7 and 8 used a population size of 300. For all sequences, 2,000 iterations of our algorithm were run. The structure of 8 protein sequences can be clearly seen in Figure 10. The results are listed in Table 4.

Table 4. Comparison of our approach with the genetic algorithm (GA), ant colony optimization (ACO), Monte Carlo (MC), tabu search with genetic algorithm (GTS), immune algorithm (IA) and estimation of distribution algorithms (EDAs). Figures in bold indicate the lowest energy.

Seq.	Length	E^*	MABC	GA	ACO	MC	GTS	IA	EDAs
1	**20**	**–9**	**–9**	**–9**	**–9**	-8	**–9**	**–9**	**–9**
2	**24**	**–9**	**–9**	**–9**	**–9**	–8	**–9**	**–9**	**–9**
3	**25**	**–8**	**–8**	**–8**	**–8**	–7	**–8**	**–8**	**–8**
4	**36**	**–14**	**–14**	–12	**–14**	–12	**–14**	**–14**	**–14**
5	**48**	**–23**	**–23**	**–22**	**–23**	–18	**–23**	**–23**	**–23**
6	**50**	**–21**	**–21**	**–21**	**–21**	–19	**–21**	**–21**	**–21**
7	**60**	**–36**	**–36**	–34	–34	–31	–35	–35	–35
8	**64**	**–42**	**–42**	–37	–32	–31	–39	–39	**–42**

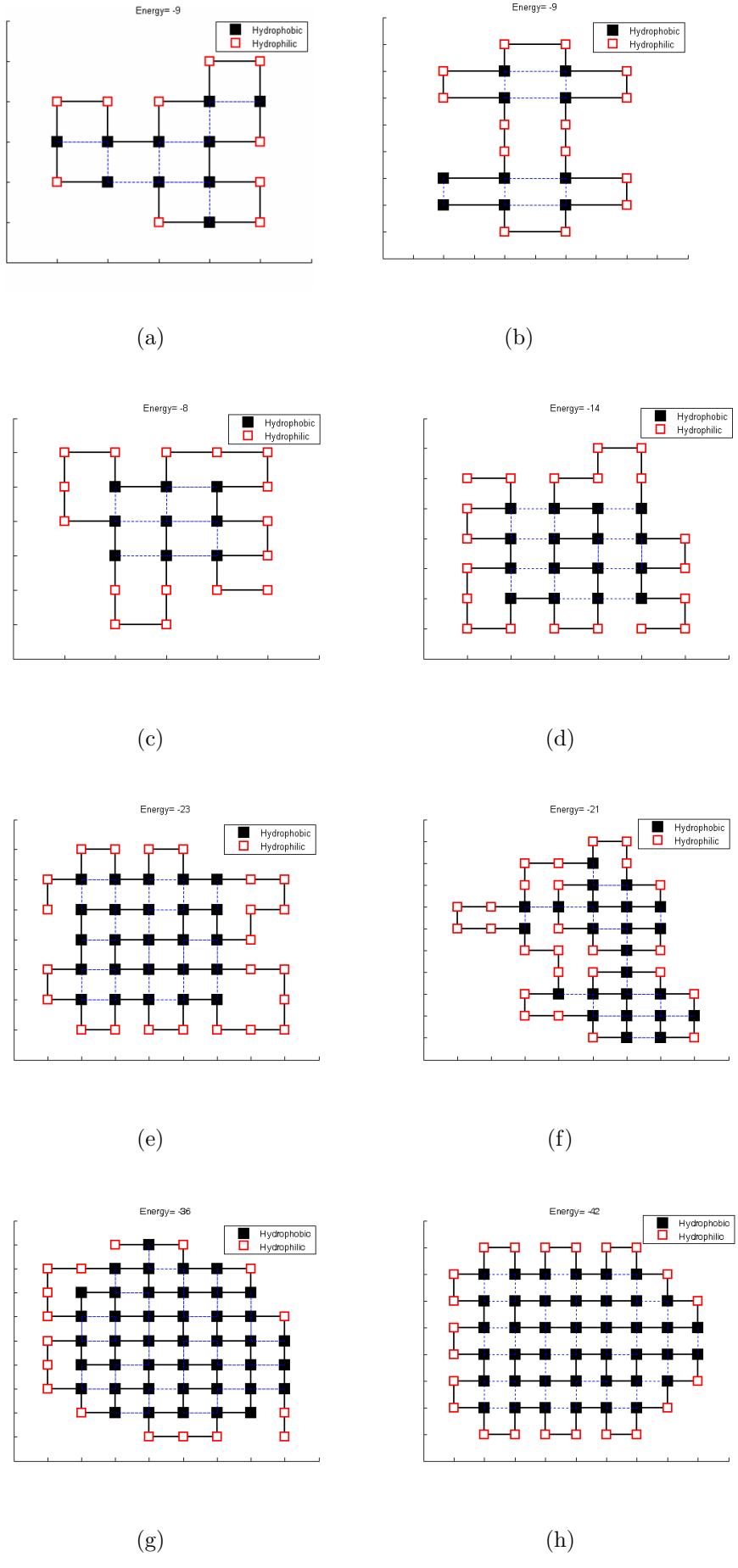


FIGURE 10. Results of the structure of 8 protein sequences

In this 3D Protein Structure, our algorithm is compared with the standard genetic algorithm, backtracking-EA [31], aging-AIS [39] and ClonalgI [40]. In Table 5, the 7 chosen HP instances are standard benchmarks used to test the searching ability of the algorithms. Sequences 1 through 7 were introduced in [24]. These sequences have been used as the benchmark for the HP model.

TABLE 5. The 3D HP benchmarks

Seq.	Length	Protein Sequence	Energy (3D)
1	20	$(HP)^2PH(HP)^2(PH)^2HP(PH)^2$	–11
2	24	$H^2P^2(HP^2)^6H^2$	–13
3	25	$P^2HP^2(H^2P^4)^3H^2$	–9
4	36	$P(P^2H^2)^2P^5H^5(H^2P^2)^2P^2H(HP^2)^2$	–18
5	48	$P^2H(P^2H^2)^2P^5H^{10}P^6(H^2P^2)^2HP^2H^5$	–29
6	50	$H^2(PH)^3PH^4PH(P^3H)^2P^4(HP^3)^2HPH^4(PH)^3PH^2$	–26
7	60	$P(PH^3)^2H^5P^3H^{10}PHP^3H^{12}P^4H^6PH^2PHP$	–49

We give the structure obtained by our algorithm as follows. Fifty independent runs of the algorithms were performed. Sequence 1 through 4 and sequence 6 were out of 100 population size. Sequence 5 and sequence 7 were out of 300 population size. For all sequences, 20,000 iterations of our algorithm were run. The structure of 7 protein sequences can be clearly seen in Figure 11.

The results are listed in Table 6, which shows a performance comparison of the various existing algorithms. In Backtracking-EA [31], the experiments were done with an elitist generational EA (population size = 100, crossover rate = 0.9 and mutation rate = 0.01) using linear ranking selection ($\eta = 2.0$). A maximum number of 10^5 evaluations were enforced. The Aging-AIS used the standard parameter values $k = 10$, $dup = 2$ and $c = 0.4$, as described in [37]. B cells had the aging parameter $\tau_B = 5$, with the memory B cells $\tau_{Bm} = 10$, and a maximum number of evaluations equal to 10^5. ClonalgI used the 10 individuals in the population. The duplication rate was equal to 4, the mutation rate was equal to 0.6, and the termination criterion was 10^5 evaluations.

TABLE 6. The simulation results obtained from the proposed algorithm compared with the methods given in the literature. Figures in bold indicate the lowest energy.

Seq.	Length	E*	MABC	GA	Backtracking-EA	Aging-AIS	ClonalgI
1	20	–11	**–11**	**–11**	**–11**	**–11**	**–11**
2	24	–13	**–13**	**–13**	**–13**	**–13**	**–13**
3	25	–9	**–9**	**–9**	**–9**	**–9**	**–9**
4	36	–18	**–18**	**–18**	**–18**	**–18**	**–18**
5	48	–29	**–29**	–25	–25	**–29**	**–29**
6	50	–26	**–26**	–23	–23	–23	**–26**
7	60	–49	**–49**	–37	–39	–41	–48

In summary, Table 4 shows that better results of stimulation from the proposed approach can be achieved on 2D-square HP lattice model than other methods and that a lower fitness value can be obtained. Similarly, the approach proposed from our study can perform very well on 3D cubic HP lattice model, as indicated in Table 6.

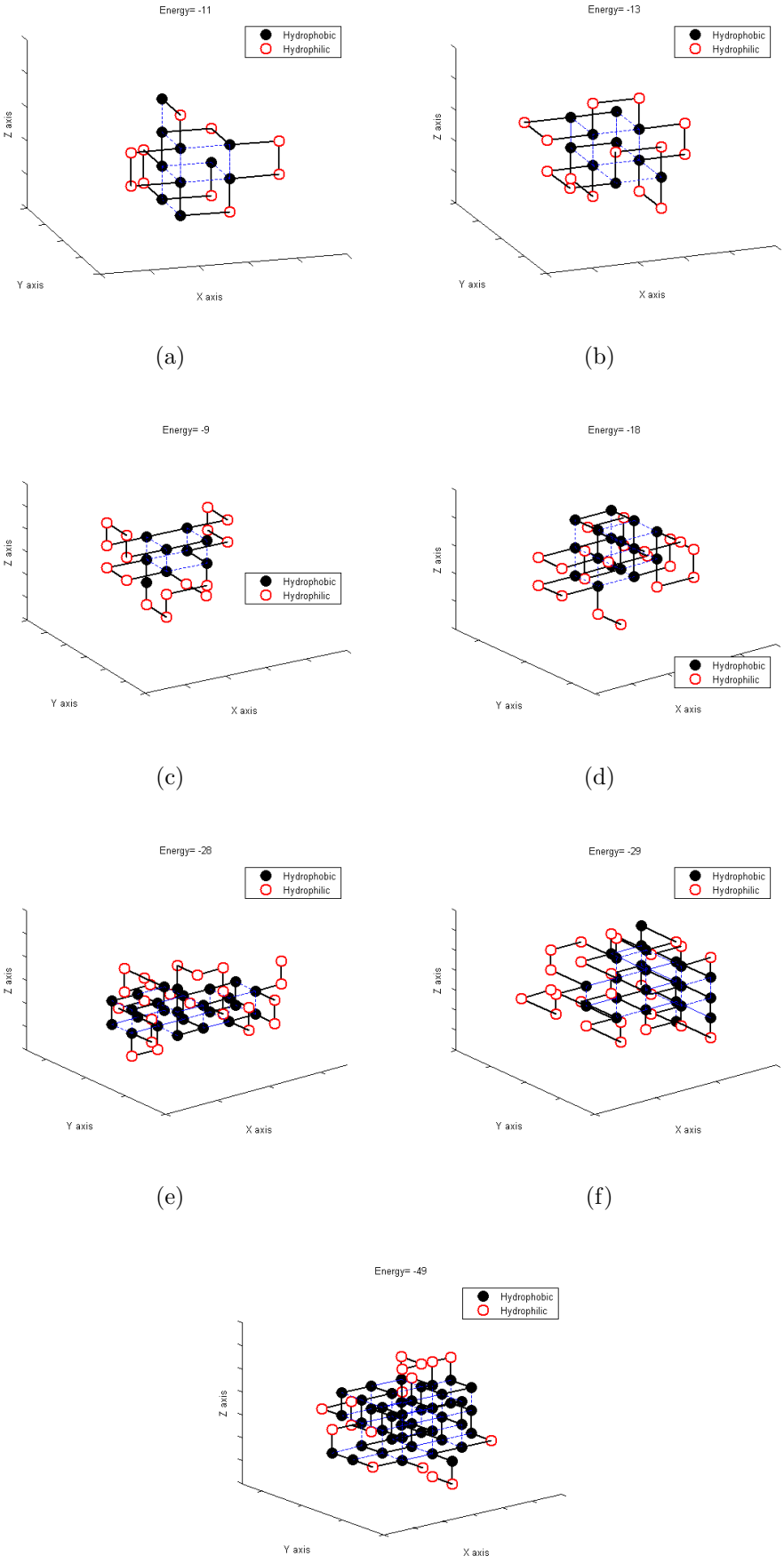


FIGURE 11. Results of the structure of 7 protein sequence

5. **Conclusions.** In this study, we present an efficient modified artificial bee colony algorithm for solving 2D/3D protein folding problem. We demonstrated that the proposed algorithm can be applied successfully to the protein folding problem based on the 2D and 3D hydrophobic-polar lattice model. The performance of the MABC algorithm is compared with genetic algorithm and other five algorithms widely used by researchers. The simulation results of the experiments show that the modified artificial bee colony algorithm can successfully be applied to the protein folding problem.

REFERENCES

[1] D. Karaboga, An idea based on honey bee swarm for numerical optimization, *Technical Report*, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005.

[2] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *IEEE International Conference on Neural Networks*, vol.4, pp.1942-1948, 1995.

[3] R. Storn and K. Price, Differential evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces, *Technical Report*, International Computer Science Institute, Berkley, 1995.

[4] D. Karaboga and B. B. Akay, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing*, vol.8, pp.687-697, 2008.

[5] D. Karaboga and B. B. Akay, A comparative study of artificial bee colony algorithm, *Applied Mathematics and Computation*, vol.214, pp.108-132, 2009.

[6] D. Karaboga and C. Ozturk, A novel clustering approach: Artificial bee colony (ABC) algorithm, *Applied Soft Computing*, vol.11, no.1, pp.652-657, 2010.

[7] D. Karaboga and B. B. Akay, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, *Foundations of Fuzzy Logic and Soft Computing, LNCS*, vol.4529, pp.789-798, 2007.

[8] D. Karaboga, B. B. Akay and C. Ozturk, Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks, *Modeling Decisions for Artificial Intelligence, LNCS*, vol.4617, pp.318-329, 2007.

[9] D. Karaboga and B. B. Akay, An artificial bee colony (ABC) algorithm on training artificial neural networks, *Proc. of the 15th IEEE Signal Processing and Communications Applications*, Eskisehir, Turkey, pp.1-4, 2007.

[10] N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters, *Journal of The Franklin Institute*, vol.346, pp.328-348, 2009.

[11] A. Singh, An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem, *Applied Soft Computing*, vol.9, pp.625-631, 2009.

[12] So much more to know, *Science*, vol.309, no.5731, pp.78-102, 2005.

[13] M. Seringhaus and M. Gerstein, Chemistry Nobel rich in structure, *Science*, vol.315, pp.40-41, 2007.

[14] L. Bragg, *The Development of X-Ray Analysis*, G. Bell, London, U.K., 1975.

[15] T. L. Blundell and L. H. Johnson, *Protein Crystallography*, Academic, New York, 1976.

[16] K. Wuthrich, *NMR of Proteins and Nucleic Acids*, Wiley, New York, 1986.

[17] E. N. Baldwin, I. T. Weber, R. S. Charles, J. Xuan, E. Appella, M. Yamada, K. Matsushima, B. F. P. Edwards, G. M. Clore, A. M. Gro-nenborn and A. Wlodawar, Crystal structure of interleukin 8: Sym-biosis of NMR and crystallography, *Proc. Nat. Acad. Sci.*, vol.88, pp.502-506, 1991.

[18] R. Unger and J. Moult, Genetic algorithms for protein folding simulations, *J. Molecular Biol.*, vol.231, pp.75-81, 1993.

[19] K. A. Dill, Theory for the folding and stability of globular proteins, *Biochemistry*, vol.24, pp.1501-1509, 1985.

[20] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni and M. Yannakakis, On the complexity of protein folding, *J. Comp. Biol.*, vol.5, pp.409-422, 1998.

[21] B. Berger and T. Leighton, Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete, *J. Comp. Biol.*, vol.5, pp.27-40, 1998.

[22] W. E. Hart and S. Istrail, Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal, *J. Comp. Biol.*, vol.3, pp.53-96, 1996.

[23] W. E. Hart and S. Coveney, Lattice and off-lattice side chain models of protein folding: Linear time structure prediction better than 86% of optimal, *J. Comp. Biol.*, vol.4, pp.241-259, 1997.

[24] T. Z. Jiang, Q. H. Cui, G. H. Shi and S. D. Ma, Protein folding simulations of the hydrophobic-hydrophilic model by combining Tabu search with genetic algorithms, *J. Chem. Phys.*, vol.119, pp.4592-4596, 2003.
[25] A. Shmygelska, R. A. Hernández and H. H. Hoos, An ant colony optimization algorithm for the 2D HP protein folding problem, *Proc. of the 3rd International Workshop on Ant Algorithms, LNCS*, vol.2463, pp.40-53, 2002.
[26] A. Shmygelska and H. H. Hoos, An improved ant colony optimization algorithm for the 2D HP protein folding problem, *Advances in Artificial Intelligence, LNCS*, vol.2671, pp.400-417, 2003.
[27] A. Shmygelska and H. H. Hoons, An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem, *BMC Bioinformatics*, vol.6, no.30, pp.1-22, 2005.
[28] V. Cutello, G. Nicosia, M. Pavone and J. Timmis, An immune algorithm for protein structure prediction on lattice models, *IEEE Transactions on Evolutionary Computation*, vol.11, pp.101-117, 2007.
[29] R. Santana, P. Larrañaga and J. A. Lozano, Protein folding in simplified models with estimation of distribution algorithms, *IEEE Transactions on Evolutionary Computation*, vol.12, pp.418-438, 2008.
[30] J. Hockenmaier, A. K. Joshi and K. A. Dill, Routes are trees: The parsing perspective on protein folding, *Proteins*, vol.66, pp.1-15, 2007.
[31] C. Huang, X. Yang and Z. He, Protein folding simulations of 2D HP model by the genetic algorithm based on optimal secondary structures, *Computational Biology and Chemistry*, pp.137-142, 2010.
[32] A. Shmygelska and H. H. Hoons, An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem, *BMC Bioinformatics*, vol.6, no.30, pp.1-22, 2005.
[33] C. Cotta, Protein structure prediction using evolutionary algorithms hybridized with backtracking, *Proc of the 7th International Work-Conference on Artificial and Natural Neural Networks, LNCS*, vol.2687, pp.321-328, 2003.
[34] V. Tereshko, Reaction-diffusion model of a honeybee colony's foraging behaviour, *Parallel Problem Solving from Nature PPSN VI, LNCS*, vol.1917, pp.807-816, 2000.
[35] V. Tereshko and T. Lee, How information mapping patterns determine foraging behaviour of a honeybee colony, *Open Systems and Information Dynamics*, vol.9, no.2, pp.181-193, 2002.
[36] V. Tereshko and A. Loengarov, Collective decision-making in honeybee foraging dynamics, *Computing and Information Systems*, vol.9, 2005.
[37] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Pub. Co., Boston, MA, USA, 1989.
[38] F. M. Liang and W. H. Wong, Evolutionary Monte Carlo for protein folding simulations, *Journal of Chemical Physics*, vol.115, pp.3374-3380, 2001.
[39] V. Cutello, G. Morelli, G. Nicosia and M. Pavone, Immune algorithms with aging operators for the string folding problem and the protein folding problem, *Evolutionary Computation in Combinatorial Optimization, LNCS*, vol.3448, pp.80-90, 2005.
[40] C. P. de Almeida, R. A. Gonçalves and M. R. Delgado, A hybrid immune-based system for the protein folding problem, *Evolutionary Computation in Combinatorial Optimization, LNCS*, vol.4446, pp.13-24, 2007.
[41] H.-Y. Chung, S.-C. Ou and C.-Y Chung, Molecular simulation via Lyapunov principle and NURBS curves, *International Journal of Innovative Computing, Information and Control*, vol.5, no.8, pp.2277-2290, 2009.