Protein Folding in Hydrophobic-Polar Lattice Model: A Flexible Ant Colony Optimization Approach

Xiao-min Hu[1,2], Jun Zhang[1,2,*], Jing Xiao[1], and Yun Li[3]

[1] Department of Computer Science, Sun Yat-sen University, Guangzhou, 510275, P. R. China
[2] Key Laboratory of Digital Life (Sun Yat-sen University), Ministry of Education, Guangzhou 510275, P. R. China
[3] Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow G12 8LT, Scotland, U.K.

Abstract: This paper proposes a flexible ant colony (FAC) algorithm for solving protein folding problems based on the hydrophobic-polar square lattice model. Collaborations of novel pheromone and heuristic strategies in the proposed algorithm make it more effective in predicting structures of proteins compared with other state-of-the-art algorithms.

Key words— ant colony optimization, protein folding, hydrophobic-polar (HP) model, protein structure prediction

* Corresponding Author; Tel: +86-2084038836; Fax: +86-2084038836; E-mail: junzhang@ieee.org

1. INTRODUCTION

As the functions of proteins are closely related to their structures, the most important step in protein analysis is to find out the structure of the proteins [1-5]. In nature, proteins fold spontaneously to their native structures very fast (on a time scale of milliseconds) in an aqueous solution [6,7]. Although the current experimental techniques, such as X-ray crystallography, nuclear magnetic resonance (NMR), protein engineering, mass spectrometry, fluorescence resonance energy transfer (FRET), and atomic force microscopy (AFM) [8-11] can be used to produce reflections of the protein, they are expensive and time-consuming. More importantly, each experimental technique has different application restrictions. Since Anfinsen *et al*. [12,13] made a remarkable discovery that many simple protein sequences had a unique native structure, which only depended on their sequences, computational methods for predicting structures of proteins become feasible. Computational methods can be applied in a much wider range, and the accuracy of predictions is much higher than that achieved by experimental methods. Basing on the sequence to predict the structure of the protein forms a kind of protein folding problems (PFPs).

A common accepted hypothesis of protein folding is that protein sequences fold into structures with the equilibrium minimum free energy (MFE) state (the thermodynamic hypothesis) [14-17]. Thus, the aim of solving a PFP is to find the protein folding conformation that satisfies the MFE state. The hydrophobic-hydrophilic (or hydrophobic-polar, HP) square lattice model [18-23] is an important abstract computational model for constructing the MFE conformations of proteins. It places the protein sequence in a square lattice, where each amino acid in the protein occupies a square of the lattice without

overlaps. Amino acids in the protein are divided into two kinds in this model. One is hydrophobic amino acids (H), and the other is hydrophilic amino acids (P). Two adjacent hydrophobic amino acids in the lattice, which are not consecutive in the protein sequence, have an energy value -1 and the adjacency is termed an H-H bond. The objective of the PFP is to find the protein conformation with the maximum number of those H-H bonds.

PFPs are nondeterministic polynomial (NP) complete problems [24], so they can not be solved in polynomial time complexity so far, except for some special protein sequences. Various computational methods have been applied in PFPs with the HP square lattice model, such as genetic algorithms (GAs) [25-29], Monte Carlo (MC) algorithms [30-35], Immune system (IM) algorithms [36,37] and ant colony optimization (ACO) [38-41] etc. Since the first ant algorithm – ant system (AS) was proposed by Dorigo [42], successors such as the elitist ant system (EAS), max-min ant system (MMAS), and ant colony system (ACS) [43,44] etc. at last form an optimization paradigm which is termed ACO. ACO algorithms have been successfully applied to a wide range of application problems, such as the vehicle routing problem (VRP) [45], the job shop scheduling problem (JSP) [46], and the water distribution system (WDS) [47], etc. The characteristics of ants' heuristic constructions and the pheromone laying method have also been shown to be promising in folding proteins in the lattice model by [38-41].

This paper proposes a simple but effective ant colony algorithm for solving PFPs, which is called the 'flexible ant colony (FAC) algorithm'. It has four special mechanisms, including the path construction, the path retrieval, the pheromone attraction, and the folding heuristics. These novel mechanisms make it behave differently from previous ant algorithms [38-41] for

solving PFPs in the HP square lattice model.

The ants of the FAC algorithm aim to find a ‘conformation path’ of a protein sequence in the lattice. Pheromones are deposited on the virtual connections between adjacent squares in the lattice. The virtual connections are the four directions up (U), down (D), left (L), and right (R) from the square pointing to its adjacent squares. Such pheromone laying approach is similar to that in a traveling salesman problem (TSP) [42], when the squares in the lattice are regarded as cities and the connections between squares are considered as arcs. It is different from the existing ant algorithms [38-41] proposed for solving PFPs, whose pheromones are on the three relative folding directions of the protein. The advantage of the proposed pheromone laying way is that the pheromones are fixed in the lattice so that ants can sense the global protein folding situations directly from the lattice board.

A protein sequence in the HP square lattice model is a string of hydrophobic (H) and polar (P) amino acids. Artificial ants place the amino acids one by one. If the surrounding lattice squares of an amino acid are all occupied, the next amino acid cannot be placed. Such situation is termed ‘stagnation’. Then the path retrieval strategy should be applied. As all ants start to construct the folding path from the center of the lattice, the pheromone value on the virtual connections between the squares that the ant just passed is decreased, aiming to add diversity for ants to choose alternative squares to place the amino acid. Such pheromone reduction method during solution construction is similar to the local pheromone update used in the ACS algorithm [43,44].

In the proposed FAC algorithm, pheromones and heuristics cooperate to construct the conformations. The heuristic information varies between the hydrophobic amino acids and

the polar amino acids, which is also different from that in [38-41]. The performance of the proposed algorithm is compared with a GA, an MC, an IA, and an ant algorithm in the literature.

The rest of the paper is constructed as follows: Section 2 presents a brief review on the PFPs with the HP model and discusses the features of protein folding conformations. Section 3 details the ants' construction behaviors of the proposed FAC algorithm. Section 4 uses the proposed algorithm to fold some benchmark protein sequences and compares the test results with some other well-known algorithms. For further analysis, this paper also tests the influences of the parameters of the FAC algorithm and highlights some prospects on enhancements. Finally, conclusions are drawn in Section 5.

2. HP SQUARE LATTICE PROTEIN FOLDING MODEL

Some benchmark instances of protein sequences in a 2-dimensional square lattice HP (2D-HP) model is listed in Table **1**, where l is the number of amino acids and E^* stands for the MFE level. The letter 'H' stands for the hydrophobic amino acid and 'P' stands for the polar amino acid, which is hydrophilic. There are 20 amino acids in nature. Using various classifications, they can be divided into acid, alkaline or neutral; positively or negatively charged or uncharged; and hydrophobic or hydrophilic, etc. To a globular protein in an aqueous solution, its hydrophilic amino acids tend to be on the surface of the globule as they are attracted to water molecules (note that the environment inside cells is primarily water). The hydrophobic amino acids are repelled by water, so that most of them gather inside the globular protein to form a core except for some special hydrophobic regions on the surface of

the protein.

Some conformations of the 2D-HP folding structures of the same protein sequence with 48 amino acids are presented in Figure (**1**). For the square lattice HP model, the best conformation is judged by the number of hydrophobic-hydrophobic (H-H) bonds that hydrophobic amino acids are adjacent on the lattice, but not consecutive in the sequence. The number of H-H bonds in each conformation in Figure (**1**) is 23 (i.e., the number of the dashed lines in the first conformation), which forms the MFE state with $E^* = -23$. It can be seen that the hydrophobic amino acids do form a core inside the protein conformation, while the polar amino acids are surrounding the core and their placements are quite flexible.

Although the square lattice model is highly abstract from the real protein folding model, some special conformations can reflect a possible secondary structure of a protein. Figure (**2**) shows the special 2D-HP conformations and the corresponding three-dimensional protein structures of an α-helix and β-sheets. The 3-dimensional structure of a specific protein sequence is unique, but as we can see in Figure (**1**), there may be several equivalent conformations by the same protein sequence. Therefore, the HP model is too simple to reflect the real protein structure completely. However, it is already a very challenging computational model.

3. ANT COLONY SEARCH IN LATTICES

Given a two-dimensional square lattice board, the PFP is to place the protein sequence in the lattice to form a self-avoiding path. The mission of an ant colony is to discover a path, which maximizes the number of H-H bonds.

3.1 Path Construction

In order not to violate the region of the lattice, each ant starts to build the path from the middle of the protein sequence in the center of the lattice. For a protein sequence with n amino acids, which are denoted as $\{s_0, s_1, \ldots, s_{n-1}\}$ ($s_j \in \{\mathrm{P},\mathrm{H}\}, j = 0,...,n-1$), each ant starts from the two horizontal squares in the middle of an $(n+2)\times(n+2)$ lattice board as depicted in Figure (**3**). The squares in the lattice board are indexed from the top left corner to the bottom right corner, by the numbers from 0 to $(n+2)^2-1$. Therefore, the two start squares with indices as $\left(\lfloor n/2\rfloor+1\right)\cdot\left(n+2\right)+\lfloor n/2\rfloor$ and $\left(\lfloor n/2\rfloor+1\right)\cdot\left(n+2\right)+\lfloor n/2\rfloor+1$ are termed the 'left start square' and the 'right start square' respectively. The two squares are colored in the middle of the lattice shown in Figure (**3**). The amino acid $s_{\lfloor n/2\rfloor}$ is placed in the left start square while the amino acid $s_{\lfloor n/2\rfloor+1}$ is placed in the right start square. The sub-protein sequence $\{s_0,...,s_{\lfloor n/2\rfloor}\}$ that is built from the left start square is denoted as the 'left path', while the $\{s_{\lfloor n/2\rfloor+1},...,s_{n-1}\}$ is the 'right path'. Then an ant randomly chooses to go a step on the left part or on the right part of the protein sequence. After several construction steps, a protein conformation is built, similar to the dashed lines in Figure (**3**). The squares that have been passed by the ant cannot be visited again by the same ant.

There are two advantages of indexing the squares in the lattice. One is that the coordinates of the squares are now one-dimensional. The other is that the four adjacent squares are convenient to obtain. For example, when an ant is now in square i as shown in Figure (**4**), which is not on the border of the lattice, the ant can go left (L), right (R), up (U), or down (D) to the corresponding square with index as i-1, i+1, i-(n+2), or i+(n+2). As the ant has passed the right square, it can only choose one of the other three movements.

3.2 Path Retrieval

If the ant has visited all the adjacent squares when placing a non-ending amino acid s_j $(j \neq 0, n-1)$, the protein cannot fold any more. Such situation is termed 'stagnation'. In this case, the folding needs to be retrieved. Consider an ant has constructed a sub-sequence $\{s_{left},\ldots, s_{startL}, s_{startR},\ldots, s_{right}\}$, where *left* is the index of the left most amino acid, *right* is the index of the right most amino acid, $startL = \lfloor n/2 \rfloor$, $startR = \lfloor n/2 \rfloor + 1$. For a 'right' retrieval, when *startR<right*, a random index *j* is selected as

$$j = rand \ \% \ (right - startR) + startR \tag{1}$$

where *rand* is a random non-negative integer number. The amino acids from s_{j+1} to s_{right} are released as not been constructed by the ant and the corresponding squares in the lattice are set vacant. On the other hand, a 'left' retrieval point *j* is selected as

$$j = rand \ \% \ (startL - left) + left + 1 \tag{2}$$

when *startL>left*. The amino acids from s_{left} to s_{j-1} are released and the corresponding squares are set vacant.

Although stagnation occurs on the right side of the protein, it does not mean that only the right side of the protein is to be retrieved, because some stagnation situations cannot be cleared by simply retrieving the side where the stagnation happens. Figure (**5**) illustrates two stagnation situations on the right path. The hollow beads stand for the left start amino acid and the right start amino acid, while the solid squares denote amino acids on the left path and the triangles are amino acids on the right path. In the example presented in Figure (**5**a), the stagnation can be released by the right retrieval when $j = 21$ and the ant is to go upward. However, in Figure (**5**b), the stagnation cannot be released by performing right retrieval but

only left retrieval. So the Boolean values *RightRetrievalBool* and *LeftRetrievalBool* are used to judge such situations to make sure that the retrieval in the same direction cannot be performed twice consecutively.

If the stagnation happens on the right side of the protein, we term the retrieval procedure as 'RightSideRetrieval', while the procedure for the left stagnation is termed 'LeftSideRetrieval'. Figure (**6**) illustrates the pseudo-code of the above process. The functions 'RightRetrievalSequence( )' and 'LeftRetrievalSequence( )' respectively perform the right/left retrieval. Take Figure (**5**b) as an example. The stagnation happens on the right path, so that the 'RightSideRetrieval' procedure is invoked. As *startR*=21, *right*=24, and *RightRetrievalBool*=false, a random integer *j* is generated by (1). Suppose *j*=22. Then the 'RightRetrievalSequence' function is invoked, so that the amino acids from 23 to 24 are released. The 'LeftRetrievalBool' and the 'RightRetrievalBool' are set as false and true respectively. It is known that this could not help to clear the stagnation. The construction of the path continues, until the stagnation happens again. Suppose the sequence changes to be 2 to 24, which is not drawn in the figure. At that time, the 'RightSideRetrieval' procedure is invoked again. As *RightRetrievalBool* is true now, it can only perform 'LeftRetrievalSequence( )' to release some amino acids on the left path. The stagnation can be cleared if *j* equals any integer from 5 to 20.

3.3 Pheromone Attraction

Pheromones are released on the directed virtual connections between adjacent squares, which are denoted as τ_{id}, where $i = 0,1,2,...,(n+2)^2 - 1$ and $d = \mathrm{L,R,U,D}$. Note that the protein sequence cannot exceed the lattice board, thus the width of the board must be bigger

than the length of the protein.

1) Local Pheromone Update

As all ants start from the same left and right start squares in the lattice, an effective method to avoid early convergence is to remove some pheromones between the two adjacent squares as (3)

$$\tau_{id} \leftarrow \delta \cdot \tau_{id}, \tag{3}$$

where *d* is the movement that the ant will go to place the next amino acid, *i* is the index of the current square in which the ant locates. $\delta = (m-1)/m < 1$ is a 'local evaporation rate', and *m* is the number of ants. If the pheromone value is smaller than $\tau_{\min}$, the value is reset to $\tau_{\min}$, which is the lower boundary of the pheromone value.

2) Global Pheromone Update

Once all ants have constructed their protein folding paths, the pheromones on all connections between adjacent squares are evaporated as (4)

$$\tau_{id} \leftarrow \rho \cdot \tau_{id}, \tag{4}$$

where ρ is a 'global evaporation rate', $i = 0,1,2,...,(n+2)^2 - 1$ and $d = \mathrm{L,R,U,D}$. Then the best path found in the current iteration is reinforced by increasing the amount of pheromone as described as (5)

$$\tau_{i'd} \leftarrow \tau_{i'd} + \varepsilon / (-E^*_{\min}), \tag{5}$$

where $i' \in$ {the squares that the iteration's best ant passed}, *d* is the movement that the ant went to the adjacent square from square i', ε is the maximum number of H-H bonds in the current iteration, $E^*_{\min} < 0$ is the approximation of the MFE of the protein in the square lattice HP model.

3.4 Heuristics for Folding

While pheromones are the means for keeping the historical memories, heuristics are the strategies for the current selection. Different from the heuristic information in [38-41] where only hydrophobic amino acids are considered, this paper takes into account both the heuristic information for hydrophobic amino acids and polar amino acids.

1) Heuristic for hydrophobic (H) amino acids

The goal for PFPs is to find the minimum energy conformation, which is reflected by the number of H-H bonds. Hence, if a conformation can yield more H-H bonds, it should have a higher probability to be constructed. Once the next amino acid s_j for an ant k to place is known as a hydrophobic (H) amino acid, the heuristics for it is determined by

$$\eta_{jd} = h_{jd} + 1, \tag{6}$$

where h_{jd} is the number of the new obtained H-H bonds by placing the amino acid s_j in the adjacent square. d is the ant's movement.

Figure (**4**) illustrates an ant that is currently located in square i with an amino acid s_{j+1}. The next step it chooses to place an amino acid s_j. The slashed squares are the feasible locations for placing s_j. For each of the feasible locations, check the adjacent squares for the number of potential new H-H bonds.

2) Heuristic for polar (P) amino acids

If the next amino acid s_j to be placed is a polar amino acid, the heuristic value is the summation number of the vacant squares and polar amino acids in the neighborhood of the possible location of the next amino acid plus one, which is given by

$$\eta_{jd} = v_{jd} + h'_{jd} + 1 \tag{7}$$

where v_{jd} and h'_{jd} are the numbers of vacant squares and polar amino acids in the neighborhood of the possible locations of s_j. Note that consecutive polar amino acid in the protein sequence is not considered.

3.5 Implementation of the Flexible Ant Colony Algorithm for PFPs

The proportional selection method is used for each ant in the colony to choose the next step. If the ant currently locates in square *i* and the next amino acid to be placed is s_j, then the probability of selecting the feasible movement *d* is given by

$$p_d = \frac{\tau_{id} \cdot \eta_{jd}^{\beta}}{\sum_{q \in \{\text{feasible movements}\}} (\tau_{iq} \cdot \eta_{jq}^{\beta})}, \tag{8}$$

where β is the reinforcement to heuristic values.

The implementation of the FAC algorithm can be realized as follows:

Step 1: Read in the protein sequence and initialize the parameters.

Step 2: Place all ants in the left start square and the right start square in the lattice.

Step 3: All ants construct feasible folding conformations to the input protein sequence. The local pheromone update is performed after every movement of ants.

Step 4: Evaluate the constructed folding paths and select the best ant in an iteration.

Step 5: Perform a global pheromone update.

Step 6: If the terminal condition is not met, go to Step 2; else terminate the algorithm.

A more detailed flowchart of the proposed algorithm is illustrated in Figure (**7**).

4. EXPERIMENTS AND DISCUSSIONS

The benchmark instances of the HP protein folding are tabulated in Table **1**. The parameter settings for the proposed FAC algorithm are $\tau_0 = 1/3$, $\tau_{\min} = 0.05$ and $\rho = 0.9$.

For sequences No. 1~7, m = 10, $\beta = 2$. For sequence No. 8, m = 100 and $\beta = 3$. Each group of parameters has been tested 30 times independently for statistical significance. The CPU time of the algorithm was recorded on a 2.8 GHz Pentium IV PC.

4.1 Comparisons with Existing Algorithms

The performance of the proposed FAC algorithm is compared with the algorithms presented in [25], [32], [36] and [38], which are the genetic algorithm (GA) [25], the evolutionary Monte Carlo (EMC) algorithm [32], the immune algorithm (IA) algorithm [36], and the ant colony optimization (ACO) [38]. The reason for choosing these algorithms is that they are representative algorithms for solving PFPs. Table **2** compares the average performance of the IA, the ACO and the proposed FAC algorithm, in terms of the average time required (*AvgT*), the average energy evaluations (*A.E.E*), and the successful rate (%ok). Table **3** compares the best time (*BestT*), the best energy evaluations (*B.E.E*), and the best number of iterations (*B.N.I*) among the FAC, the EMC, the GA, and the IA.

In Table **2**, the values in bold denote the best results of the three algorithms. Except for sequence 1, the average function evaluations of the FAC are much smaller than those of the IA. Moreover, the FAC has successfully found the best protein conformation in all tests, while the IA only manages to solve sequence 8 with a 56.67% successful rate. Compared with the ACO, the average execution time of the FAC in obtaining the best protein for short protein sequences is not significantly longer, but it takes a shorter time for longer sequences such as 7 and 8.

In Table **3**, the best performances of the algorithms are compared. Only are the best energy evaluations to sequence 6 by the FAC slightly larger than those of the IA. It can be

seen that the FAC algorithm developed in this paper can solve the given PFPs in a very short time.

4.2 Analysis on Different Parameters Values

The influence of parameters in the FAC algorithm is tested in order to assess the best group of values for the parameters, including the number of ants *m*, the heuristic reinforcement value β, and the global pheromone evaporation rate ρ. Figure (**8**) shows the trends of different parameter values for sequences 1 to 7.

1) The Heuristic Reinforcement Value β

Fix the values of *m* and ρ. When β increases, the time needed to obtain solutions becomes shorter for sequences 1 to 5. Note that sequence 6 is distinctive in the sequences and it achieves the best result when $\beta = 1$.

2) The Pheromone Evaporation Rate ρ

If the pheromone evaporation rate ρ is about 0.9, the performance of the FAC is good in most of the test cases. Overall, the influence of ρ is not so significant as β.

3) The Number of Ants *m*

A large number of ants provide a higher insurance of finding the best conformation, but it slows down the algorithm. However, a small number of ants may induce early convergence to sub-optima. The proper number of ants generally depends upon the length of the protein sequence. For short protein sequences, *m*=10 is enough. However, for long sequences such as the one with 48 amino acids, more ants (e.g., *m*=100) are needed.

4.3 Analysis on Heuristic Information to Polar Amino Acids

In the proposed FAC algorithm, there is heuristic information for folding polar amino

acids. The performance of the FAC algorithm with or without heuristic information to polar amino acids is compared. The results are tabulated in Table **4**. With the same parameter settings, the algorithm without heuristic information to polar amino acids is slower than the one with the heuristic information in all test cases. The results demonstrate that the heuristic information proposed in this paper is effective.

5. CONCLUSIONS

This paper has presented a flexible ant colony (FAC) algorithm for protein folding problems (PFPs). This FAC algorithm is based on the 2-dimensional square lattice hydrophobic-polar (HP) model, which is a highly abstract model for protein folding structures. Ants in the FAC algorithm start from the middle of the lattice and construct the protein folding from the middle of the protein sequence. Pheromones are released to the directed virtual connections between adjacent squares in the lattice. Local pheromone update as well as global pheromone update mechanisms are also implemented. By using effective heuristic and pheromone method for selection, the proposed FAC algorithm can solve the PFP fast as shown by the experimental results. Comparison with some well-known PFP algorithms has highlighted superior performance of the proposed FAC algorithm.

ACKNOWLEDGEMENT

This work was financially supported by the National Natural Science Foundation of China under Grant No. 60573066, the Guangdong Natural Science Foundation Research under Grant No. 5003346, and the Scientific Research Foundation for the Returned Overseas

Chinese Scholars, State Education Ministry, P. R. China.

REFERENCES

[1] Chandru, V. Dattasharma, A. and Kumar, V.S.A. (**2003**) *Discrete Appl. Math.*, *127*, 145–161.

[2] Taylor, W.R. (**1986**) *J. Mol. Biol.*, *188*, 233–258.

[3] Zhao, X.M. Cheung, Y.M. and Huang, D.S. (**2005**) *Protein Pept. Lett.*, *12*, 665-669.

[4] Mount, D.W. (**2001**) *Bioinformatics: sequence and genome analysis*, 382-398.

[5] Bhasin, M. and Raghava, G.P.S. (**2004**) *Nucleic Acids Res.*, *32*, 383–389.

[6] Zaki, M.J. Jin, S. and Bystroff, C. (**2003**) *IEEE Trans. Syst. Man Cybern. B Cybern.*, *33*, 789–810.

[7] Onuchic, J.N. Wolynes, P.G. (**2004**) *Curr. Opin. Struct. Biol.*, *14*, 70-75.

[8] Chen, Y.W. Feng, D. Nie, H.F Serohijos, A.W. Sharma, S. Wilcox, K.C. Yin, S.Y and Dokholyan, V. (**2008**) *Arch. Biochem. Biophys.*, *469*, 4-19.

[9] Chen, S.S. (**2001**) *Int. J. Intell. Syst.*, *16*, 449–457.

[10] Woldawer, A. Miller, M. Jaskolski, M. Sathyanarayana, B.K. Baldwin, E. Weber, I.T. Selk, L.M. Clawson, L. Schneider, J. and Kent, S.B. (**1989**) *Science*, *245*, 616–621.

[11] Clore, G.M. and Gronenborn, A.M. (**1991**) *J. Mol. Biol.*, *221*, 47–53.

[12] Anfinsen, C.B. Haber, E. Sela, M. and White, F.H. (**1961**) *Proc. Natl. Acad. Sci. USA*, *47*, 1309–1314.

[13] Anfinsen, C.B. (**1973**) *Science*, *181*, 223–230.

[14] Unger, R. and Moult, J. (**1993**) *Bull. Math. Biol.*, *55*,1183–1198.

[15] Govindarajan, S. and Goldstein, R.A. (**1998**) *Proc. Natl. Acad. Sci. USA*, *95*, 5545–5549.

[16] Huang, D.S. and Huang, X. (**2006**) *Protein Pept. Lett.*, *13*, 985-991.

[17] Huang, D.S. Zhao, X.M. Huang, G.B. and Cheung, Y.M. (**2006**) *Pattern Recogn.*, *39*, 2293–2300.

[18] Dill, K.A. (**1985**) *Biochemistry*, *24*, 1501-1509.

[19] Guo, Y.Z. and Feng, E.M. (**2006**) *J. Chem. Phys.*, *125*, 154102.

[20] Lesh, N. Mitzenmacher, M. and Whitesides, S. (**2003**) *Proc. Annu. Int. Conf. Res. Comput. Mol. Biol.*, *1*, 188–195.

[21] Dill, K.A. Bromberg, S. Yue, K. Fiebig, K.M. Yee, D.P. Thomas, P.D. and Chan, H.S. (**1995**) *Protein Sci.*, *4*, 561–602.

[22] Hart, W.E. and Istrail, S. (**1995**) *Conf. Proc. Annu. ACM Symp. Theor. Comput.*, *1*, 157–168.

[23] Chan, H.S. and Dill, K.A. (**1994**) *J. Chem. Phys.*, *100*, 9238–9257.

[24] Crescenzi, P. Goldman, D. Papadimitriou, C. Piccolboni, A. and Yannakakis, M. (**1998**) *J. Comp. Biol.*, *5*, 423–466.

[25] Unger, R. and Moult, J. (**1993**) *J. Mol. Biol.*, *231*, 75–81.

[26] Unger, R. and Moult, J. (**1993**) *Proc. Hawaii Int. Conf. Syst. Sci.*, *1*, 715-725.

[27] Hoque, M.T. Chetty, M. and Dooley, L.S. (**2005**) *Proc. IEEE Congr. Evol. Comput.*, *1*, 259–266.

[28] Bui, T.N. and Sundarraj, G. (**2005**) *Conf. Proc. Genet. Evol. Comput.*, *1*, 385–392.

[29] Custódio, F.L. Barbosa H.J.C. and Dardenne, L.E. (**2004**) *Genet. Mol. Biol.*, *27*, 611–615.

[30] Frauenkron, H. Bastolla, U. Gerstner, E. Grassberger, P. and Nadler, W. (**1998**) *Phys. Rev. Lett.*, *80*, 3149-3152

[31] Bastolla, U. Frauenkron, H. Gerstner, E. Grassberger, P. and Nadler, W. (**1998**) *Protein Struct. Funct. Genet.*, *32*, 52–66.

[32] Liang, F. and Wong, W.H. (**2001**) *J. Chem. Phys.*, *115*, 3374–3380.

[33] Hsu, H.P. Mehra, V. Nadler, W. and Grassberger, P. (**2003**) *J. Chem. Phys.*, *118*, 444.

[34] Ramakrishnan, R. Ramachandran, B. and Pekny, J.F. (**1997**) *J. Chem. Phys.*, *106*, 2418–2425.

[35] Zhang, G.Z. and Huang, D.S. (**2004**) *J. Comput. Aided Mol. Des.*, *18*, 797-810.

[36] Cutello, V. Nicosia, G. Pavone, M. and Timmis, J. (**2007**) *IEEE Trans. Evol. Comput.*, *11*, 101–117.

[37] Cutello, V. Nicosia, G. and Pavone, M. (**2004**) *Proc. IEEE Congr. Evol. Comput.*, *1*, 1074–1080.

[38] Shmygelska, A. and Hoos, H.H. (**2005**) *BMC Bioinformatics*, *6:30*.

[39] Chu, D. Till, M. and Zomaya, A. (**2005**) *Proc. IEEE Int. Parallel Distr. Process. Symp.*, *1*, 1–7.

[40] Shmygelska, A. and Hoos, H.H. (**2003**) *Lect. Notes Artif. Intell.*, *2671*, 400-417.

[41] Shmygelska, A. Aguirre-Hernandez, R. and Hoos, H.H. (**2002**) *Lect. Notes Comput. Sci.*, *2463*, 40–52.

[42] Dorigo, M. and Gambardella, L.M. (**1997**) *IEEE Trans. Evol. Comput.*, *1*, 53–66.

[43] Dorigo, M. and Stützle, T. (**2004**) *Ant Colony Optimization*, 1-319.

[44] Dorigo, M. Maniezzo, V. and Colorni, A. (**1996**) *IEEE Trans. Syst. Man Cybern. B*

Cybern., *26*, 29–41.

[45] Gambardella, L.M.; Taillard, E.D.; and Agazzi, G. (**1999**) in *New Ideas in Optimization* (Corne, D.; Dorigo, M. and Glover, F., Ed). pp. 63-76. London McGraw Hill, UK.

[46] Zhang, J. Hu, X.M. Tan, X. Zhong, J.H. and Huang, Q. (**2006**) *Trans. Inst. Meas. Contr.*, *28*, 1–16.

[47] Zecchin, A.C. Simpson, A.R. Maier, H.R. and Nixon, J.B. (**2005**) *IEEE Trans. Evol. Comput.*, *9*, 175–191.

Figure Captions

Table Captions

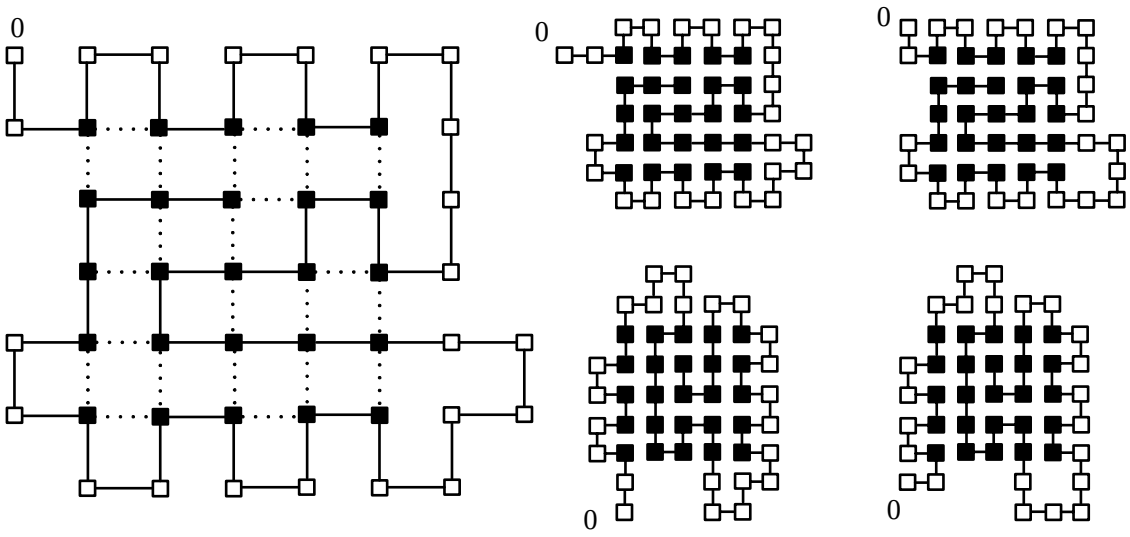


Figure (**1**) Some Conformations of Sequence 8 (Length = 48)

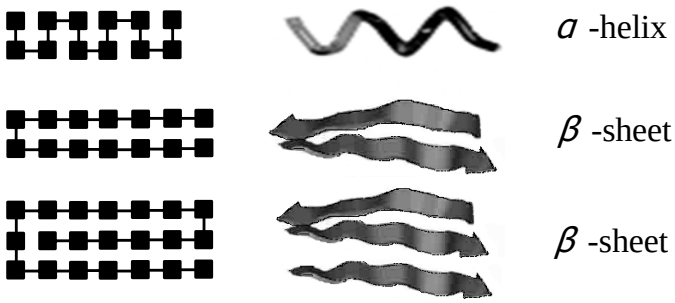


Figure (**2**) Special HP Conformations and the Secondary Structures of Protein Sequence

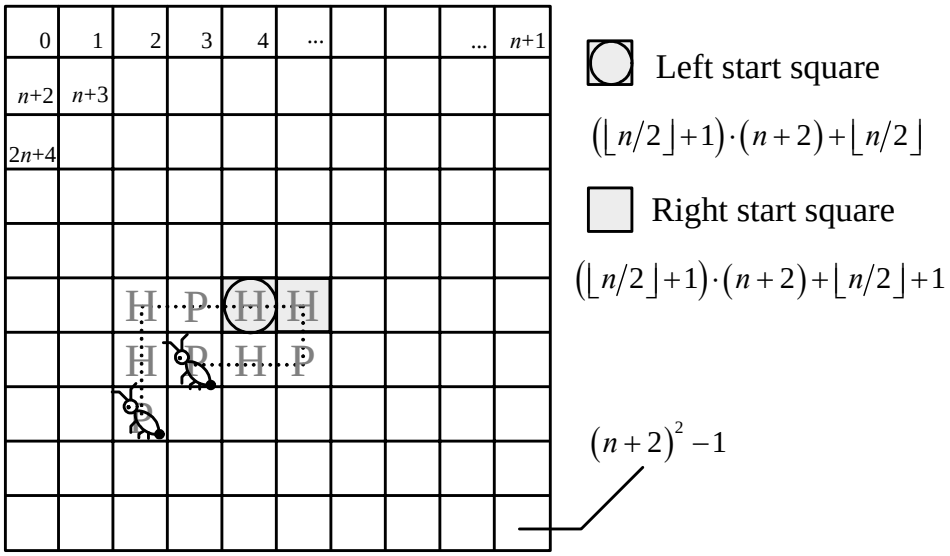


Figure (**3**) The Lattice Board for a Protein with *n* Amino Acids

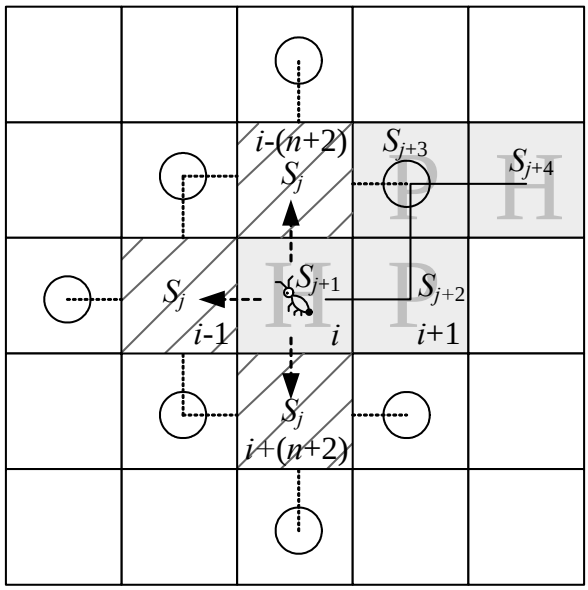


Figure (**4**) An Ant Chooses a Step to Go

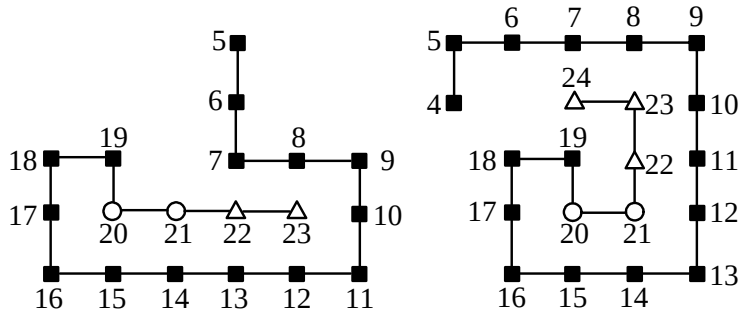


Figure (**5**) Examples of Stagnation

```
/* startL = ⌊n/2⌋, startR = ⌊n/2⌋+1
   left : the constructed left end, right: the constructed right end */
Procedure RightSideRetrieval({s_left,…, s_startL, s_startR,…, s_right})
   If startR < right && RightRetrievalBool == false
      j = rand % (right – startR) + startR;
      RightRetrievalSequence({s_left,…, s_startL, s_startR,…, s_right}, j);
      LeftRetrievalBool = false;
      RightRetrievalBool = true;
   Else If startL > left
      j = rand % (startL – left) + left +1;
      LeftRetrievalSequence({s_left,…, s_startL, s_startR,…, s_right}, j);
      RightRetrievalBool = false;
   End

Procedure LeftSideRetrieval({s_left,…, s_startL, s_startR,…, s_right})
   If startL > left && LeftRetrievalBool == false
      j = rand % (startL – left) + left +1;
      LeftRetrievalSequence({s_left,…, s_startL, s_startR,…, s_right}, j);
      LeftRetrievalBool = true;
      RightRetrievalBool = false;
   Else If startR < right
      j = rand % (right – startR) + startR;
      RightRetrievalSequence({s_left,…, s_startL, s_startR,…, s_right}, j);
      LeftRetrievalBool = false;
   End
```

Figure (**6**) Outline of the Retrieval Process

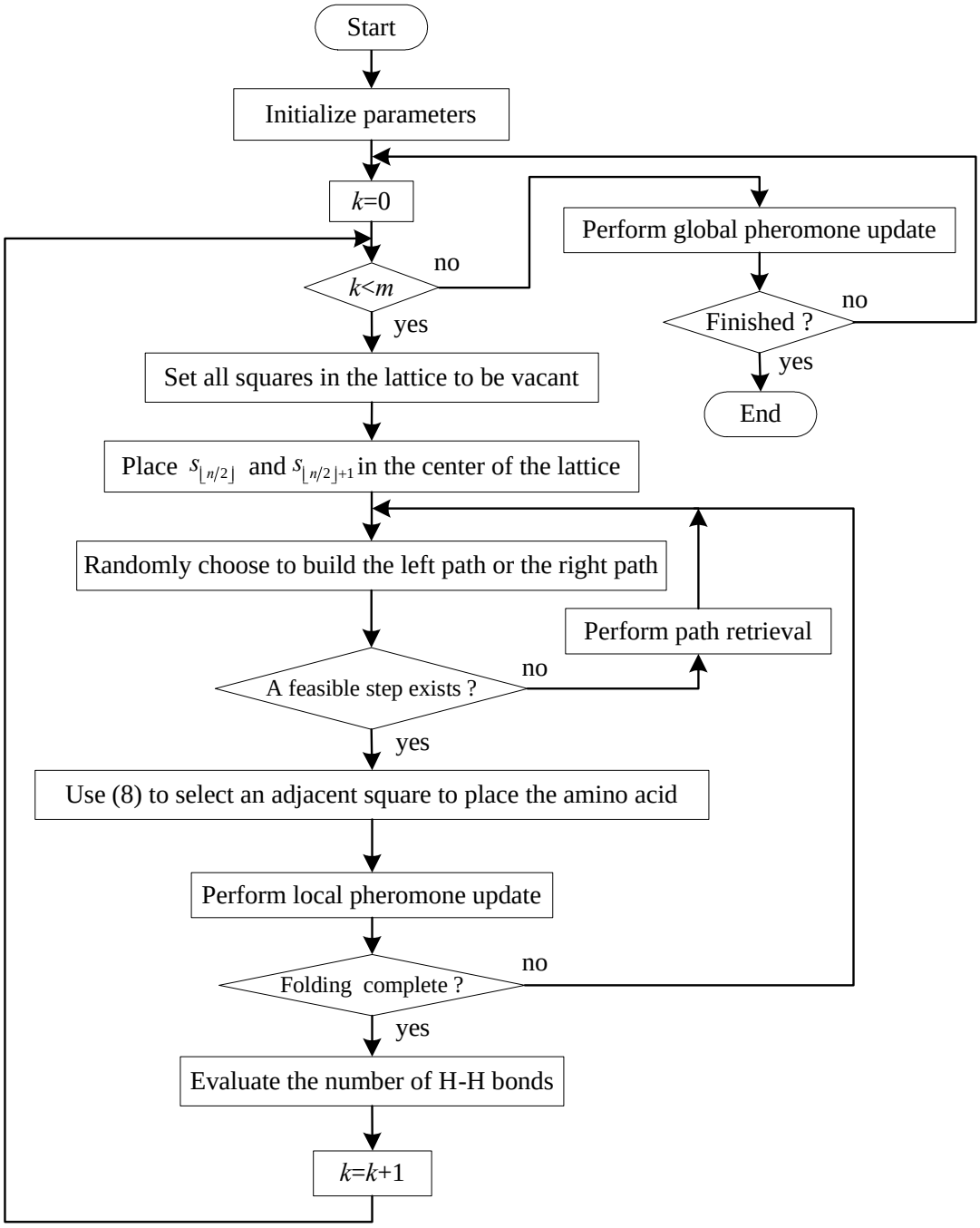
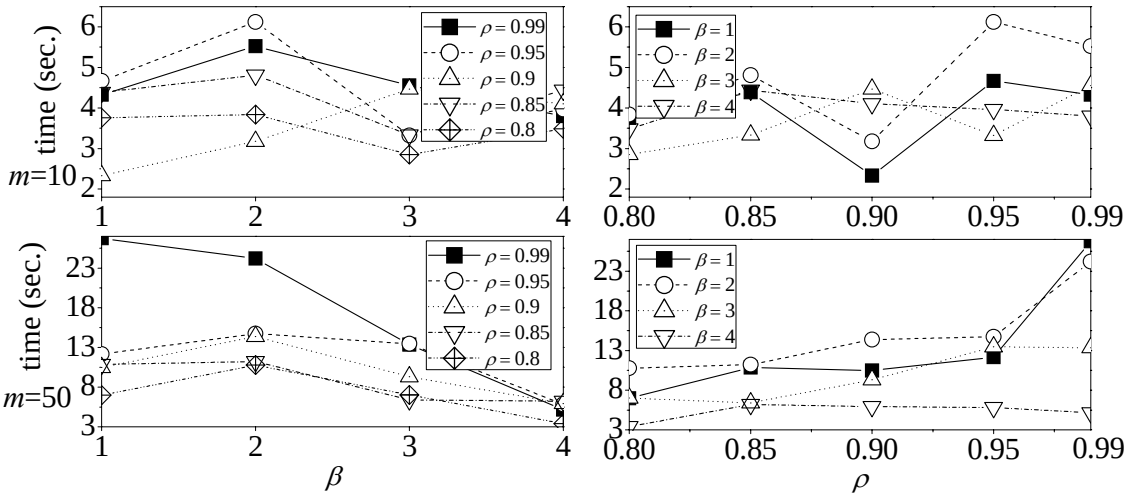
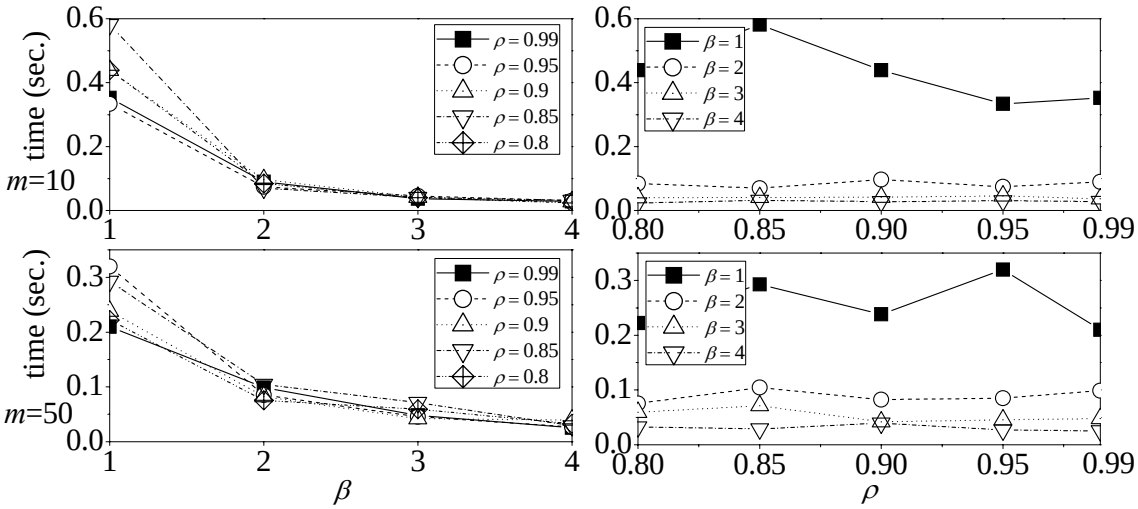


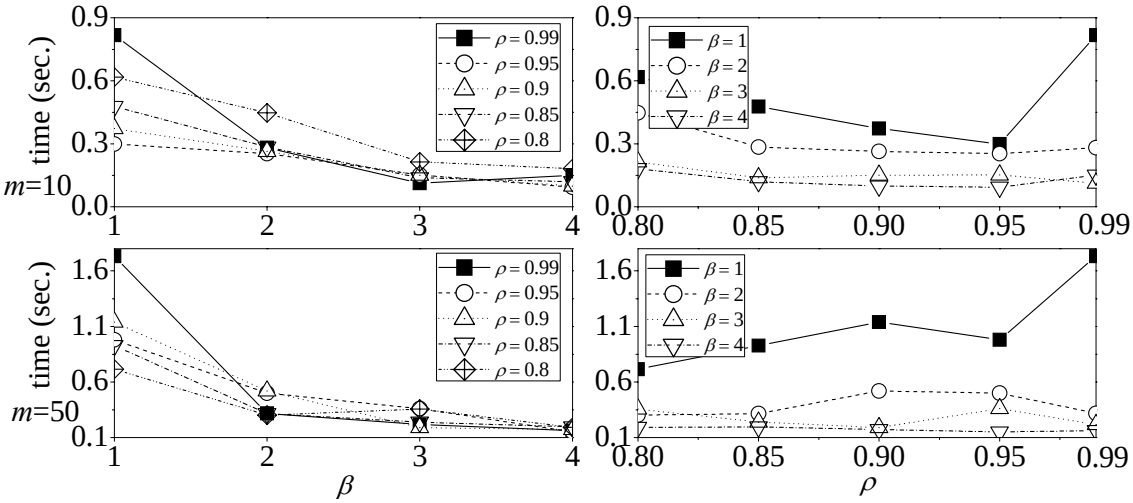
Figure (**7**) Flowchart of the FAC Algorithm



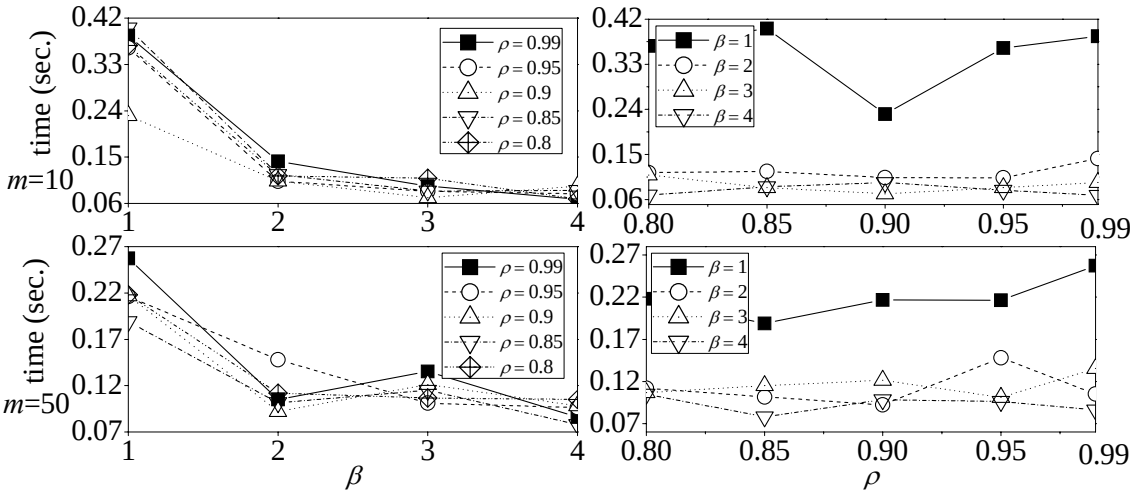
(a) Sequence 1 with m = 10 and m = 50



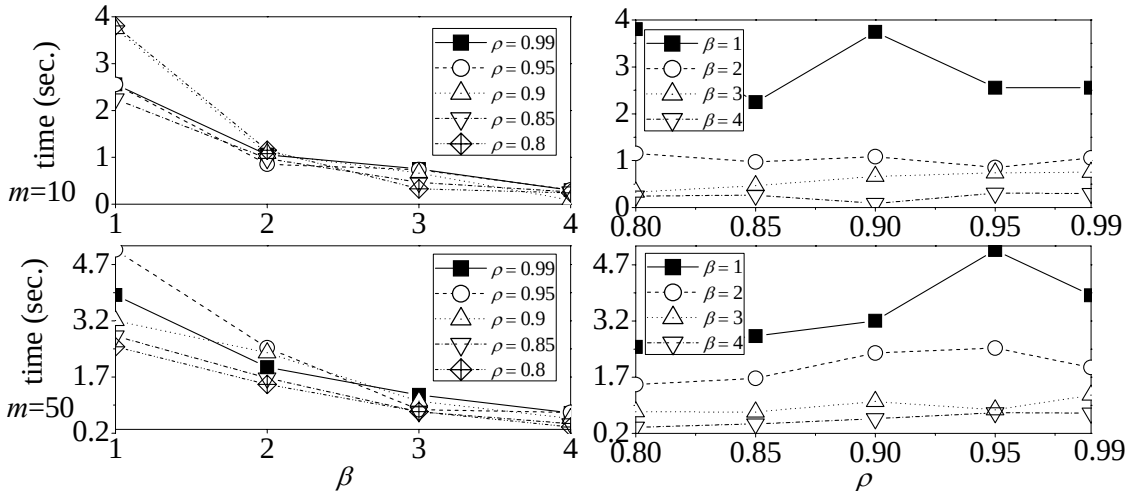
(b) Sequence 2 with m = 10 and m = 50



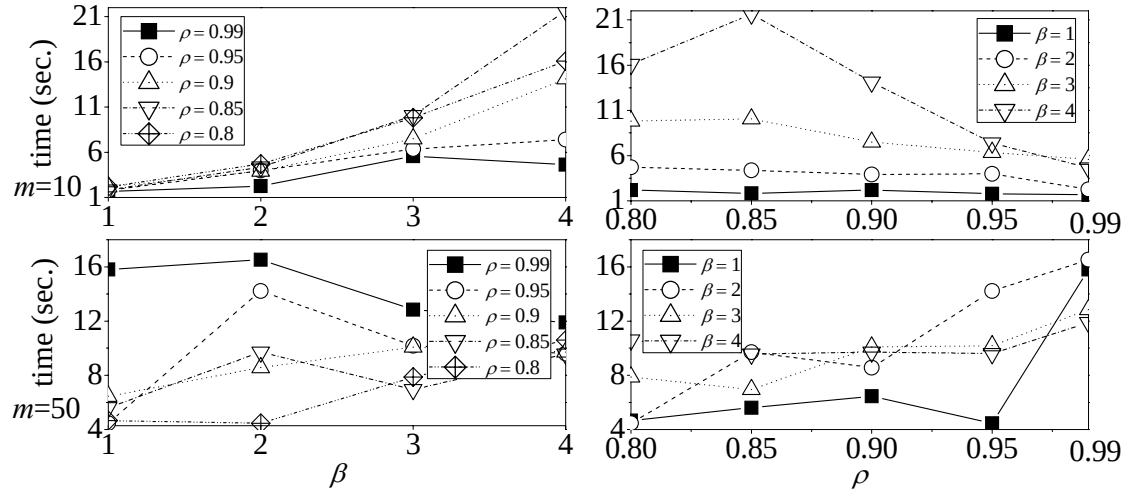
(c) Sequence 3 with m = 10 and m = 50



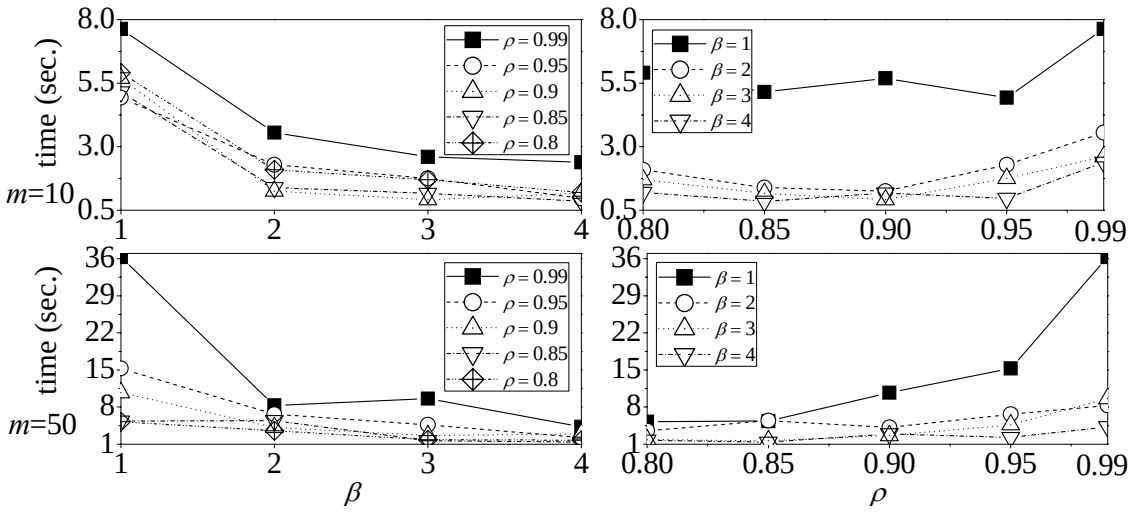
(d) Sequence 4 with m = 10 and m = 50



(e) Sequence 5 with m = 10 and m = 50



(f) Sequence 6 with m = 10 and m = 50



(g) Sequence 7 with m = 10 and m = 50

Figure (**8**) Analysis on the FAC Algorithm with Different Parameter Values

Table **1** Standard HP Benchmarks for 2-D Square Lattice

No.	*l*	E^*	Protein Sequence
1	18	-9	PHPPHPHHHPHHPHHHHH
2	18	-8	HPHPHHHPPPHHHHPPHH
3	20	-10	HHHPPHPHPHPPHPHPHPPH
4	20	-9	HPHPPHHPHPPHPHHPPHPH
5	24	-9	HHPPHPPHPPHPPHPPHPPHPPHH
6	25	-8	PPHPPHHPPPPHHPPPPHHPPPPHH
7	36	-14	PPPHHPPHHPPPPPHHHHHHHHPPHHPPPPHHPPHPP
8	48	-23	PPHPPHHPPHHPPPPPHHHHHHHHHHPPPPPPHHPPHHPPHPPHHHHH

Table **2** Comparisons of Average Performances for the 2D-HP Problems

No.	*l*	E^*	FAC			IA		ACO	
			AvgT(sec.)	*A.E.E*	%ok	*A.E.E*	%ok	*AvgT*(sec.)	%ok
1	18	-9	3.17703	115384	100	**69210**	100	--	--
2	18	-8	0.0967667	**3149**	100	41724.2	100	--	--
3	20	-10	0.264167	**8107**	100	18085.8	100	--	--
4	20	-9	**0.103667**	**2981**	100	23710	100	**< 1 sec.**	100
5	24	-9	1.2833	**32159**	100	69816.7	100	**< 1 sec.**	100
6	25	-8	3.90027	**93883**	100	269513.9	100	**< 1 sec.**	100
7	36	-14	**1.25527**	**18683**	100	2032504	100	4 sec	100
8	48	-23	**28.922**	**331103**	100	6403985	56.67	1 min.	100

-- The corresponding values are unavailable in the reference.

Table **3** Comparisons of Best Performances for the 2D-HP Problems

No.	*l*	E^*	FAC			EMC	GA	IA
			BestT(sec.)	*B.E.E*	*B.N.I*	*B.E.E*	*B.E.E*	*B.E.E*
4	20	-9	0.015	**169**	17	9374	30492	1925
5	24	-9	0.078	**1703**	171	6929	30491	2479
6	25	-8	0.234	5463	547	7202	20400	**4212**
7	36	-14	0.031	**234**	24	12447	301339	43416
8	48	-23	0.797	**9102**	92	165791	126547	37269

Table **4** Comparisons on Whether Using Heuristic Information to Polar Amino Acids

No.	*l*	E^*	FAC (use)			FAC (not use)		
			AvgT(sec.)	*A.E.E*	%ok	*AvgT*(sec.)	*A.E.E*	%ok
1	18	-9	**3.17703**	**115384**	100	7.69107	272580	100
2	18	-8	**0.0967667**	**3149**	100	0.181733	5903	100
3	20	-10	**0.264167**	**8107**	100	0.3943	11820	100
4	20	-9	**0.103667**	**2981**	100	0.116833	3271	100
5	24	-9	**1.2833**	**32159**	100	1.5146	36870	100
6	25	-8	**3.90027**	**93883**	100	4.14727	95673	100
7	36	-14	**1.25527**	**18683**	100	2.3422	33789	100
8	48	-23	**28.922**	**331103**	100	334.755	3756947	100

A list of Abbreviations:

HP	Hydrophobic-polar
NMR	Nuclear magnetic resonance
FRET	Fluorescence resonance energy transfer
AFM	Atomic force microscopy
PFP	Protein folding problem
MFE	Minimum free energy
H	Hydrophobic amino acid
P	Polar amino acid
H-H	Hydrophobic-hydrophobic
NP	nondeterministic polynomial
GA	Genetic algorithm
MC	Monte Carlo
ACO	Ant colony optimization
IA	Immune algorithm
FAC	Flexible ant colony
U	up
D	down
L	left
R	right
TSP	Traveling salesman problem
ACS	Ant colony system
2D-HP	2-dimensional hydrophobic-polar
AS	Ant system
EAS	Elitist ant system
MMAS	Max-min ant system
VRP	Vehicle routing problem
JSP	Job shop scheduling problem
WDS	Water distribution system
n	Number of amino acids in the sequence
E^*	MFE level
τ_{ij}	Pheromone value
$\tau_{\min}$	Lower bound of the pheromone value
δ	Local pheromone evaporation rate
m	Number of ants
η_{ij}	Heuristic value
β	Heuristic reinforcement value
ρ	Global pheromone evaporation rate
AvgT	Average time required
A.E.E	Average energy evaluations
%ok	Successful rate
BestT	Best time
B.E.E	Best energy evaluations
B.N.I	Best number of iterations