IEEJ TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING
IEEJ Trans 2010; **5**: 459–466
Published online in Wiley InterScience (www.interscience.wiley.com). DOI:10.1002/tee.20559

Paper

A Hybrid EDA for Protein Folding Based on HP Model

Benhui Chen*, Non-member
Jinglu Hu*[a], Member

Protein structure prediction (PSP) is one of the most important problems in computational biology. This paper proposes a novel hybrid estimation of distribution algorithm (EDA) to solve the PSP problem on HP model. First, a composite fitness function containing the information of folding structure core (H-core) is introduced to replace the traditional fitness function of HP model. The proposed fitness function is expected to select better individuals for the probabilistic model of EDA. Second, local search with guided operators is utilized to refine the found solutions for improving the efficiency of EDA. Third, an improved backtracking-based repairing method is proposed to repair invalid individuals sampled by the probabilistic model of EDA. It can significantly reduce the number of backtracking searching operation and the computational cost for a long-sequence protein. Experimental results demonstrate that the proposed method outperform the basic EDA method. At the same time, it is very competitive with other existing algorithms for the PSP problem on lattice HP models. © 2010 Institute of Electrical Engineers of Japan. Published by John Wiley & Sons, Inc.

Keywords: protein structure prediction (PSP), estimation of distribution algorithms (EDAs), local search, guided operator, backtracking

Received 26 March 2009; Revised 10 June 2009

1. Introduction

Protein structure prediction (PSP) is one of the most important problems in computational biology. A protein is a chain of amino acids (also called as residues) that folds into a specific native tertiary structure under certain physiological conditions. Understanding protein structures is vital to determining the function of a protein and its interaction with DNA, RNA, and enzymes. While there are over a million known protein sequences, only a limited number of protein structures have been experimentally determined. Hence, prediction of protein structures from protein sequences using computer programs is an important step to unveil proteins' three-dimensional conformation and functions.

Because of the complexity of the PSP problem, simplified models such as Dill's HP-lattice [1] model have become the major tools for investigating the general properties of protein folding. In HP model, 20-letter alphabet of residues is simplified to a two-letter alphabet, namely H (hydrophobic) and P (polar). Experiments on small proteins suggest that the native state of a protein corresponds to a free-energy minimum. This hypothesis is widely accepted, and forms the basis for computational prediction of a protein's conformation from its residue sequence. The problem of finding such a minimum-energy configuration has been proved to be NP-complete for the bi-dimensional (2-D) [2] and tri-dimensional (3-D) lattices [3,4]. Therefore, a deterministic approach is always not practical for this problem.

Many genetic algorithm (GA) based methods have been proposed to solve the problem of structure prediction in the HP model in recent years [5–8]. Although GAs have been proposed to solve this problem, the difficulties of crossover operators to deal with this type of problem have been acknowledged [9,10]. Particularly, it has been pointed out that one-point crossover and uniform crossover do not perform well for this problem, because it often produces disruption to the structure of folding solutions. Evolutionary algorithms (EAs) that are able to learn and use the relevant interactions which may arise between the variables of the problem can perform well for this problem [11], such as the estimation of distribution algorithms (EDAs).

In the EDAs [12,13], instead of using conventional crossover and mutation operations, probabilistic models are used to sample the genetic information in the next population. The use of probabilistic models, especially, models taking into account the bivariate or multivariate dependency between variables, and allows EDAs to capture genetic tendencies in the current population effectively. In brief, these algorithms construct, in each generation, a probabilistic model that estimates the probabilistic distribution of the selected solutions. Dependency regulars are then used to generate next-generation solutions during a simulation step. It is expected that the generated solutions share a number of characteristics with the selected ones. In this way, the search leads to promising areas of the search space.

In Ref. 11, EDAs that use Markov probabilistic model or other probabilistic models were shown to outperform other population-based methods when solving the HP-model folding problem, especially for the long-sequence protein instances. But those methods have three obvious disadvantages: (i) For most long-sequence protein instances, the chance of finding the global optimum is very low, and the algorithm often needs be set by a very large generation number and population size for finding the global optimum. (ii) For some deceptive sequences, those methods can only find sub-optimum solutions. (iii) In those methods, a backtracking method is used to repair invalid individuals sampled by the probabilistic model of EDAs. For a traditional backtracking algorithm, the computational cost of the repairing procedure is very heavy for those long-sequence instances.

[a] Correspondence to: Jinglu Hu. E-mail: jinglu@waseda.jp

* Graduate School of Information, Production and Systems, Waseda, University. Hibikino 2-7, Wakamatsu-ku, Kitakyushu-shi, Fukuoka 808-0135, Japan

This paper proposes a hybrid method to solve above problems of the EDA-based method for HP-model protein folding. First, a composite fitness function containing the information of folding structure core (H-core) is introduced to replace the traditional fitness function of the HP model. The proposed fitness function is expected to select better individuals for the probabilistic model of the EDAs. It can help to increase the chance of finding the global optimum and reduce the complexity of EDA (population size and the number of generation needed). Second, local search with guided operators is utilized to refine the found solutions for improving the efficiency of EDA. The local information of solutions found so far can be helpful for exploitation, while the global information can guide the search for exploring promising areas. Local search with guided operators generates offspring through a combination of global statistical information and the location information of solutions found so far. Third, an improved backtracking-based repairing method is proposed to repair invalid individuals sampled by the probabilistic model of EDAs for the long-sequence protein instances. The traditional backtracking repairing procedure will waste a lot of computational time for searching the invalid closed areas of the folding structure. In the improved method, to avoid entering invalid closed areas, a detection procedure for feasibility is introduced when selecting directions for the residues in the backtracking searching procedure. It can significantly reduce the number of backtracking searching operations and the computational cost for the long protein sequences.

The rest of the paper is organized as follows. In Section 2, we give a brief overview of protein HP model and the EDAs. In Section 3, we formulate the proposed hybrid EDA for HP-model protein folding. It includes the proposed composite fitness function and local search with guided operators. In Section 4, we formulate the improved backtracking repairing algorithm for invalid solutions. Section 6 presents the experiment results of the proposed method. Finally, the conclusions and further work directions are introduced.

2. Protein HP Model and EDAs

2.1. Protein folding and HP model Under specific conditions, a protein sequence folds into a unique native 3-D structure. Each possible protein fold has an associated energy. The thermodynamic hypothesis states that the native structure of a protein is the one for which the free energy achieves the minimum. Based on this hypothesis, many methods have been proposed to search for the protein native structure by defining an approximation of the protein energy and utilizing the optimization methods. These approaches mainly differ in the type of energy approximation employed and in the characteristics of the protein modeling.

In the HP model, a protein is considered as a sequence $S \in \{H, P\}^+$, where H represents a hydrophobic residue and P represents a hydrophilic or polar residue. The HP model restricts the space of conformations to self-avoiding paths on a lattice in which vertices are labeled by the residues.

Given a pair of residues, they are considered neighbors if they are adjacent either in the chain (connected neighbors) or in the lattice but not connected in the chain (topological neighbors). Let ε_{HH} denote the interaction energy between topological neighbors of two H residues, ε_{PP} for two P residues, ε_{HP} for an H residue and a P residue. An energy function is defined as the total energy of topological neighbors with $\varepsilon_{HH} = -1$ and $\varepsilon_{PP} = \varepsilon_{HP} = 0$. The HP problem is to find the folding conformation that minimizes the total energy $E(x)$. Figure 1 shows the graphical representation of a possible configuration for the sequence $HPHPPHHPPHPHPH$ in the 2-D HP model; hydrophobic residuals are represented by black beads and polar residuals by white beads. The energy that the HP model associates with this configuration is -6.

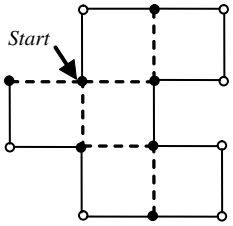

Fig. 1. One possible configuration of the sequence $HPHPP$ $HHPPHPHPH$ in 2-D HP model. There are six HH topological neighbors (represented by broken lines)

Although more complex models have been proposed, the HP model remains a focus of research in computational biology as well as chemical and statistical physics. By varying the energy function and the bead sequence of the chain (the primary structure), effects on the native state structure and the kinetics (rate) of folding can be explored, and this may provide insights into the folding of real proteins. In particular, the HP model has been used to investigate the energy landscapes of proteins, i.e. the variation of their internal free energy as a function of conformation. In evolutionary computation, the model is still employed because of its simplicity and its usefulness as a testbed for new evolutionary optimization approaches [11].

2.2. Estimation of distribution algorithms EDAs were introduced in the field of evolutionary computation in [12]. In EDAs, there are neither crossover nor mutation operators. Instead, the new population of individuals is sampled from a probability distribution, which is estimated from a database that contains the selected individuals from the previous generation. Thus, the interrelations between the different variables that represent the individuals are explicitly expressed through the joint probability distribution associated with the individuals selected at each generation. In order to explain the behavior of this heuristic, a common outline for all EDAs is listed as follows:

1. Generate the first population of M individuals and evaluate each of them. Usually, this generation is made assuming a uniform distribution on each variable.
2. N individuals are selected from the set of M, following a given selection method.
3. A one-dimensional (size of the individual) probabilistic model that shows the interdependencies among the variables is induced from the N selected individuals.
4. Finally, a new population of individuals is generated based on the sampling of the probability distribution learnt in the previous step.
5. Steps of 2 to 4 are repeated until some stop criterion is met (e.g., a maximum number of generations, a homogeneous population, or no improvement after a certain number of generations).

EDAs can be seen as a development of GAs. By recombining a subset of selected solutions, GAs are able to process the information learned during the search, and to orient the exploration

to promising areas of the search space. Nevertheless, it has been proved that GAs experience limitations in their capacity to deal with problems where there are complex interactions between different components of the solutions. In these scenarios, EDAs can exhibit a better performance [14,15].

3. Proposed Hybrid EDA for Protein Folding Based on HP Model

3.1. Problem representation for EDA In the algorithm of protein folding optimum, one of the important problems is how to present a specific conformation. To embed a hydrophobic pattern $S \in \{H, P\}^+$ into a lattice, we have three methods: Cartesian coordinate, internal coordinate, and distance matrix [16]. Krasnogor *et al*. [16] performed an exhaustive comparative study using EAs with relative and absolute directions. The experimental results show that relative directions almost always outperform absolute directions over square and cubic lattices, while absolute directions have better performances when facing triangular lattices. Experimental evidence suggests internal coordinates with relative directions should be used. However, in general, it is difficult to assess the effectiveness of direction encoding on an EA's performance.

In this paper, we use the representation of internal coordinates with relative direction, the position of each residue depending upon the previous move. Relative direction representation presents the direction of each residue relative to the next turn direction of the main chain. This representation can reduce the direction number of each position. For 2-D HP model, the set of direction is left, right, and forward (L, R, F). And it is left, right, forward, up, and down, (L, R, F, U, D) for the 3-D HP model. For example, by relative direction representation, the representation of the protein structure shown in Fig. 1 is $s = (RFRRLLRRFRLR)$.

It is to be noted that the backward direction is not used, because the backward direction will cause overlap in this representation. Thus, this representation can reduce the position collision to a certain degree to guarantee the self-avoiding walk folding procedure. There are other advantages of the relative direction representation. One is that the sequence conformation can be presented as one-dimensional array. The most important is that the change of a start direction will not influence the structure of another part in the sequence.

3.2. The probabilistic model of EDA It is very important for EDAs to select an appropriate probabilistic model according to the given application problem. The probabilistic model is represented by conditional probability distributions for each variable and estimated from the genetic information of selected individuals in the current generation. Therefore, the type of probabilistic model also influences the number and strength of the interactions learned by the model.

In Ref. 11, three probabilistic models for EDAs are proposed to solve the HP-model problem: k-order Markov model, tree model, and mixtures of the tree model. In practice, we find that the k-order Markov model is an appropriate probabilistic model for the HP-model problem, where $k > 0$ is a parameter of the model. It can effectively embody the self-avoiding folding characteristics of the HP-model problem, because it is assumed that positions of adjacent residues are related in the protein folding procedure.

The k-order Markov model can encode the dependencies between the move of a residue and the moves of the previous residues in the sequence, and this information can be used in the generation of solutions. It is described as follows: The joint probability mass function of X is denoted as $p(\mathrm{X} = X)$ or $p(X)$. And use $p(\mathrm{X}_i = x_i \,|\mathrm{X}_j = x_j)$ or the simplified form $p(x_i \,|x_j)$ to denote the conditional probability distribution of $\mathrm{X}_i = x_i$ given $\mathrm{X}_j = x_j$.

In the k-order Markov model, the value of the variable X_i depends on the values of the previous k variables. The joint probability distribution can be factorized as follows:

$$p_{MK}(X) = p(x_1, \ldots, x_{k+1}) \prod_{i=k+2}^{n} p(x_i | x_{i-1}, \ldots, x_{i-k}) \qquad (1)$$

Since the structure of the Markov model is given, it can be used to construct the probabilistic model through computing the marginal and conditional probabilities of the set of selected individuals and to sample the new generation. To sample a new solution, first, variables in the factor $(x_1, \ldots, x_{k+1})$ are generated, and the rest of variables are sampled according to the order specified by the Markov factorization.

3.3. Proposed composite fitness function In order to increase the chance of finding the global optimum and reduce the complexity of the EDA (population size and the number of generation needed), a composite fitness function containing the information of the folding structure core is introduced to replace the traditional fitness function of the HP model.

It is well known that the energy potential in the HP model reflects the fact that hydrophobic residues have a propensity to form a hydrophobic core. The H's (hydrophobic residues) form the protein core and the P's (hydrophilic or polar residues) tend to remain in the outer surface. As shown in Fig. 2, the inner kernel, called the H-core [17], is compact and mainly formed of H's while the outer kernel consists mostly of P's. The H-core center is called HCC. The H-core is a rectangle-like area in the 2-D lattice and a cube-like space in the 3-D lattice. The coordinates of HCC can be calculated by the following equations:

$$x_{HCC} = \frac{1}{n_H} \sum_{i=1}^{n_H} x_i, \quad y_{HCC} = \frac{1}{n_H} \sum_{i=1}^{n_H} y_i$$
$$z_{HCC} = \frac{1}{n_H} \sum_{i=1}^{n_H} z_i \qquad (2)$$

where n_H is the sum of hydrophobic residues in solution, and x_i, y_i, and z_i (for 3-D HP model) are the coordinates of hydrophobic residue position in the lattice. We can calculate the number of H's in inner kernel H-core (denoted as $N_{HC}(x)$) by searching the surrounding rectangular area (cube space for 3-D HP model) of HCC.

The number of H's in the inner kernel H-core is an important characteristic for the folding solution. It also reflects the optimum degree of the solution. In the practice we find that, for two solutions

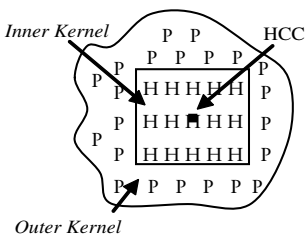

Fig. 2. The H-core of protein folding structure in 2-D HP model

with same basic HP-model energy $E(x)$ (defined by the number of topological neighbor residues in the lattice), the solution with a bigger H-core has more similarity to the optimum solution, and it also has more biological significance. So, we introduce a novel composite fitness function containing the information of H-core for the k-order Markov EDA:

$$Fit_{cp} = \omega(-E(x)) + (1-\omega)N_{HC}(x) \tag{3}$$

where $E(x)$ is the total energy of the interaction between topological neighbor residues of the HP model ($\varepsilon_{HH} = -1$, $\varepsilon_{PP} = \varepsilon_{HP} = 0$). $N_{HC}(x)$ is the number of H's in inner kernel H-core. ω is weight parameter of the fitness function, and we always take $\omega > 0.5$ because the interaction energy $E(x)$ is the dominant characteristic of protein folding solution.

3.4. Local search with guided operators An efficient EA should make use of both the local information of solutions found so far and the global information about the search space. The local information of solutions found so far can be helpful for exploitation, while the global information can guide the search for exploring promising areas. The search in EDAs is mainly based on the global information, but local search is an exploration method based on local information. Therefore, it is worthwhile investigating whether combining local search with EDA could improve the performance of the EDA.

Local search with a set of guided operators is implemented in the proposed hybrid EDA. Some of these operations have been utilized as mutations in the previous GA and ant colony optimizations studies of protein folding [7]. But in this paper, we call them as 'guided operators', meaning that those operations are implemented only under some special conditions.

Taking the 2-D HP model as example, the special conditions are defined as follows: (i) The guided operation should grantee the validity of the individual, i.e. it cannot produce position collision in the lattice. If we want to change some residues to other positions in the lattice, the object positions must be empty. (ii) Guided operation should follow a basic principle that make H's as near as possible to the HCC and P's far away from the HCC according to the relative position in lattice, as shown in Fig. 3.

The method of choosing individuals to implement local search is described as follows: In each iteration procedure of EDAs, use the composite fitness function (described by (3)) to sort the selection individuals. According the distribution of individuals' fitness, randomly select some individuals (the number is a certain percentage of the population) in each fitness domain to implement the local search with guided operators.

EDAs extract globally statistical information from the previous search and then build a probabilistic model for modeling the distribution of the best solutions visited in the search space. However, the information of the locations of the best individual solutions found so far is not directly used for guiding further search. Local search with a guided operator generates offspring through a combination of global statistical information and the location information of solutions found so far. The resultant solution can (hopefully) fall in or close to a promising area, which is characterized by the probabilistic model.

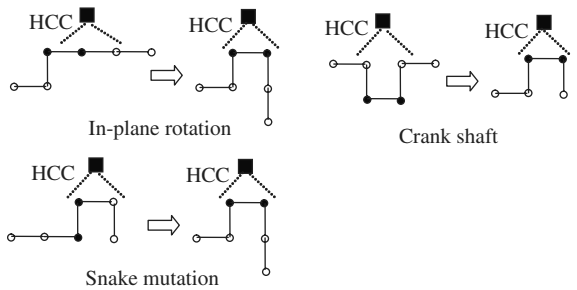

Fig. 3. The guided operators for local search

4. Improved Backtracking-Based Repairing Method

4.1. Disadvantage of traditional backtracking-based method In Ref. 8, a backtracking algorithm was introduced to repair the positional collisions. It utilizes backtracking strategy to search the feasible positions for collision residues in the folding procedure. This method has been shown to be a simple and efficient means of positional collision repairing for protein folding. But in practice, we found that the computational cost of repairing is very high for long-sequence instances of more than 50 residues.

For a long-sequence protein, there are many closed areas in the 2-D folding procedure (or closed spaces in the 3-D circumstance). Taking the 2-D circumstance as example, as shown in Fig. 4(a), we assume that there is a closed area formed by residues 1 to n. If the folding procedure selects right (R) as the next direction for the $n+1$ residue, it will enter the closed area. Thus, even if the size of this closed area cannot satisfy the length of the remaining residues (called as an invalid closed area), the traditional backtracking-based method will still search all empty positions in the closed area by the backtracking operation. This will lead to a large wastage of computational time. According to our experiment, this phenomenon takes place with a high probability in repairing procedures for long-sequence proteins.

4.2. Improved method To solve the above problem, in the improved method, a detection for feasibility is introduced. The detection procedure is implemented before selecting the direction for the next residue to avoid entering an invalid closed area. The main idea of the proposed detection procedure is described as follows:

1. The current boundaries in the lattice are defined as shown in Fig. 4(a); the scale of the boundary coordinates is larger than the current filled area and will change with the current folding procedure. They can be used to check whether the folding procedure enters a closed area. If the detection meets the current boundaries, the folding procedure will not enter a closed area.
2. Before selecting a certain direction for next residue, the folding procedure utilizes a search approach, similar to the Floodfill strategy, to detect empty positions connected to this direction, i.e. those empty positions that could be arrived through this direction.

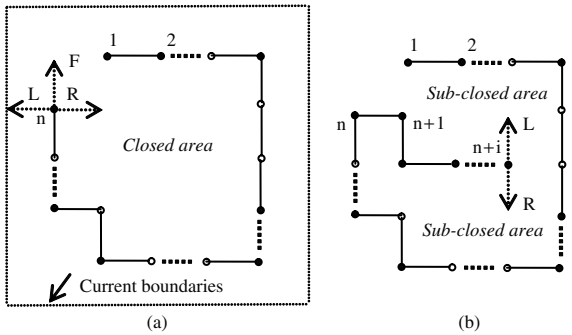

Fig. 4. (a) Illustration of closed-area detection. (b) The situation that need to implement backtracking

```
(1)  ↑ Detect-fea (↓ ⟨λ⟩, ↓ s:MOVE[]):bool.
(2)    Calculate current boundaries according to s
(3)    Set label for every positions in s (i.e. not empty).
(4)    if Feasible(s :: ⟨λ⟩) then
(5)      counter = 0 and set an empty queue Q
(6)      Add ⟨λ⟩ to the end of Q
(7)      while Q is not empty do
(8)        x=first element of Q
(9)        if position x is unlabeled
(10)         Set label for position x
(11)         counter = counter + 1
(12)       endif
(13)       if (position x meet the current boundaries) or
       (counter is larger than the length of remain residues)
(14)         Return TRUE
(15)       endif
(16)       Remove the first element of Q
(17)       if west neighbor of x is unlabeled
(18)         Set label for west − x
(19)         Add west − x to the end of Q
(20)       endif
(21)       Check and process other three (five for 3D) neighbor
       positions of x using similar strategies Step (17)-(20)
(22)     endwhile
(23)   endif
(24)   Return FALSE
```

Fig. 5. The pseudocode of the detection procedure

3. The detected direction could be chosen for the next residue only when the following condition is achieved: the detection procedure meets the current boundaries or the number of detected empty positions is larger than the length of the remaining residues.

The pseudocode of the detection algorithm is shown in Fig. 5 It use a Floodfill strategy to label the empty positions connected with the detected direction. λ is a table containing the allowed moves for each residue; thus, λ_k is a list of allowed moves for the $(k+1)$-th residue. Parameter s is a partial conformation involving $|s|$ residues. The operator represents the sequence concatenation operator.

For long protein sequences, there are many invalid closed areas in the folding procedure. The improved method can significantly reduce the computational cost. Although the detection procedure involves some computational cost, it is far less than the cost of backtracking searching operations for invalid closed areas.

The main reason of improvement is that the proposed method can significantly reduce the number of backtracking operation. The folding procedure implements backtracking operation only under a few special circumstances. As shown in Fig. 4(b), assume the folding procedure has selected the right (R) direction for the $n+1$ residues. But at the $n+i$ position, the folding procedure produces two sub-closed areas and both are invalid closed areas for the remaining residues. The folding procedure should implement a backtracking operation under this situation. It will back to the $n+i-1$ residue and search for other possible directions.

5. Experiments

5.1. Problem benchmark For our experiments, we use the first nine instances of the *Tortilla* 2-D HP Benchmarks [17], and the last two instances are taken from [11] to test the searching capability of the proposed method. In Table I, E^* is the optimal or best known energy value, and H_i, P_i, and $(\ldots)_i$ indicate i repetitions of the relative symbol or subsequence. It is important to highlight that most randomly generated amino acid sequences do not behave like natural proteins, because the latter are products of natural selection. Likewise, most randomly generated sequences of H and P residues in the HP model do not fold to a single conformation [11].

Table I. HP instances used in the experiments

No.	Size	E^*	Sequence
s1	20	−9	$hphp_2h_2php_2hph_2p_2hph$
s2	24	−9	$h_2p_2(hp_2)_6h_2$
s3	25	−8	$p_2hp_2(h_2p_4)_3h_2$
s4	36	−14	$p_3h_2p_2h_2p_5h_7p_2h_2p_4h_2p_2hp_2$
s5	48	−23	$p_2h(p_2h_2)_2p_5h_{10}p_6(h_2p_2)_2hp_2h_5$
s6	50	−21	$h_2(ph)_3ph_4p(hp_3)_2hph_4(ph)_4h$
s7	60	−36	$p_2h_3ph_8p_3h_{10}php_3h_12p_4h_6ph_2php$
s8	64	−42	$h_{12}(ph)_2(p_2h_2)_2p_2h(p_2h_2)_2p_2h$ $(p_2h_2)_2p_2(hp)_2h_{12}$
s9	85	−53	$h_4p_4h_{12}p_6(h_{12}p_3)_3h(p_2h_2)_2p_2hph$
s10	100	−48	$p_6hph_2p_5h_3ph_5ph_2p_4h_2p_2h_2ph_5ph_{10}$ $ph_2ph_7p_{11}h_7p_2hph_3p_6hphh$
s11	100	−50	$p_3h_2p_2h_4p_2h_3(ph_2)_3h_2p_8h_6p_2h_6$ $p_9hph_2ph_{11}p_2h_3ph_2php_2hph_3p_6h_3$

5.2. Results of proposed hybrid EDA for HP model In order to test the effects of the proposed composite fitness function and local search with guided mutation, we have implemented different experiments by using one of them independently. The composite fitness function can help to reduce the complexity of EDA; it can obtain the same results with the basic k-order Markov EDA (MK-EDA) by using a smaller population size and generation number. The local search with guided mutation can help to obtain the global optimum for some instances. But it seems that a combination of two strategies can get much better results in practice. We have investigated the performance of MK-EDA for $k \in \{2, 3, 4\}$ and found that the algorithm performs very well when $k = 3$.

In experiments of the proposed hybrid EDA, all algorithms use a population size of 2000 individuals. Truncation selection is used as the selection strategy. In this strategy, individuals are ordered by fitness, and the best $T * PopSize$ are selected where T is the truncation coefficient. The parameter $T = 0.15$ is used in our algorithms. The best elitism scheme is also implemented in algorithms; the set of selected solutions in the current generation is passed to the next generation. The stop criteria considered are a maximum number of generation $G = 1000$ or the number of different individuals in the population falling below 5. For the protein instances of $s6$ to $s11$, we use the improved backtracking-based method to repair the invalid solutions.

The results of the proposed method compared to the MK-EDA for the 2-D HP model are shown in Table II. It includes the best solution and the percentage of times the best solution has been found in 100 experiments. The results of MK-EDA are also obtained by our experiments with same EDA parameters ($Pop = 2000$, $G = 1000$, and $T = 0.15$) as the proposed method. From the experimental results, we can see that the proposed method has a better chance to find a global optimum or suboptimum solution for long sequences. The MK-EDA cannot find the global optimum of the deceptive sequences and long sequences $s7$, $s9$, $s10$, and $s11$, but the proposed method can find the global optimum of the sequences $s7$, $s9$, and $s10$, and can find the second best solution for the sequence $s11$. Figure 6 shows evolution of the best fitness for one representative run of instance $S7$.

The performance of the proposed method compared to the best results achieved with other evolutionary and Monte Carlo optimization algorithms is shown in Table III (2-D HP model) and Table IV (3-D HP model). The results of the other methods are

Table II. Results of comparing with MK-EDA for 2-D HP model

		Proposed method		MK-EDA	
No.	E^*	$H(X)$	*Percentage*	$H(X)$	*Percentage*
s1	−9	−9	100	−9	100
s2	−9	−9	100	−9	100
s3	−8	−8	100	−8	100
s4	−14	−14	16	−14	5
s5	−23	−23	22	−23	7
s6	−21	−21	92	−21	57
s7	−36	−36	24	−35	12
s8	−42	−42	16	−42	4
s9	−53	−53	8	−52	3
s10	−48	−48	12	−47	4
s11	−50	−49	6	−48	2

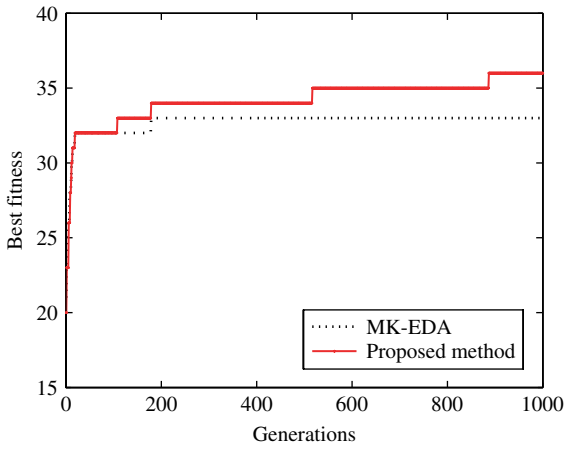

Fig. 6. Evolution of the best fitness for one representative run of instance $S7$

Table III. Results achieved by different search methods for the 2-D HP model

No.	Proposed $H(X)$	MK-EDA $H(X)$	GA $H(X)$	NewACO $H(X)$	PERM $H(X)$
s1	−9	−9	−9	−9	−9
s2	−9	−9	−9	−9	−9
s3	−8	−8	−8	−8	−8
s4	−14	−14	−14	−14	−14
s5	−23	−23	−22	−23	−23
s6	−21	−21	−21	−21	−21
s7	−36	−35	−34	−36	−36
s8	−42	−42	−37	−42	−38
s9	−53	−52		−51	−53
s10	−48	−47		−47	−48
s11	−49	−48		−47	−50

cited from [11,18].From the experimental results, we can find that none of the algorithms is able to outperform the rest of the algorithms for all instances. The Pruned-Enriched Rosenbluth method (PERM) is one of the best contenders in all cases except $s8$ in which its result is very poor. It shows that our method is very competitive with the other existing algorithms for PSP on lattice HP models. It should be noted that all fitness values of the proposed method in the compared results are calculated by basic HP-model fitness definition. The proposed composite fitness function is only used in the optimization procedure of EDA.

Table IV. Results achieved by different search methods for the 3-D HP model

No.	Proposed $H(X)$	MK-EDA $H(X)$	Hybrid GA $H(X)$	IA $H(X)$
s1	−11	−11	−11	−11
s2	−13	−13	−11	−13
s3	−9	−9	−9	−9
s4	−18	−18	−18	−18
s5	−29	−29	−28	−28
s6	−30	−29	−22	−23
s7	−49	−48	−48	−41
s8	−51	−50	−46	−42

5.3. Results of comparing computational cost

The proposed hybrid EDA is an improved method based on the MK-EDA. The detailed computational cost analysis of the MK-EDA method can be found in Ref. 11. Compared to the MK-EDA, there are three modifications in the proposed method: (i) the proposed composite fitness function; (ii) the local search with guided operations; (iii) the improved backtracking-based repairing method for the long protein instances. As far as the computational cost is concerned, modifications i and ii will produce some additional computational cost. The modification iii can significantly reduce the repairing costs for EDA invalid individuals.

To demonstrate the computational cost of the proposed hybrid EDA compared to MK-EDA, some practical experiments in 2-D are implemented. Two comparison methods are implemented with same parameters (population:1000, generation:100, the truncation selection of parameter $T = 0.15$), and same computational environment. Because there are few closed areas existing in a short protein folding, the improved backtracking-based repairing method cannot improve the EDA efficiency for short instances. In comparing experiments for the short instances of $s1$ to $s5$, the same basic backtracking repairing methods are used in the two comparison methods. For the long instances of $s6$ to $s10$, the improved backtracking-based repairing method is used in the proposed hybrid EDA.

The number of backtracking searching operation and computer CPU time are recorded. The average backtracking searching operations and the CPU times for 10 runs are shown in Table V. According to the results of the short instances of $s1$ to $s5$, we find that the local search operations and composite fitness calculation in the proposed hybrid EDA produce some additional computational costs. But it is not very serious. The results of the long instances of $s6$ to $s10$ show that the proposed repairing method can significantly reduce the repairing costs. It not only covers the additional computational costs caused by local search and composite fitness calculation but also improves the algorithm efficiency remarkably.

The backtracking searching operations of each generation for sequences $s9$ (the length is 85) and $s10$ (the length is 100) are also be counted, and shown in Figs. 7 and 8. We can find that the improved backtracking-based repairing method can significantly reduce the number of backtracking searching operation. And the longer the protein sequence length, the more remarkable the improvement achieved.

Table V. Comparing the results of improved backtracking repairing method in the 2-D HP model

No.	Size	Average-backtracking operation		Average-CPU time (h)	
		MK-EDA	Proposed	MK-EDA	Proposed
s1	20	2.1492E+4	2.1496E+4	0.2181	0.2309
s2	24	2.5715E+4	2.5715E+4	0.2659	0.2761
s3	25	2.6834E+4	2.7044E+4	0.2774	0.2805
s4	36	3.8898E+4	3.8797E+4	0.3059	0.3203
s5	48	5.2577E+5	5.2617E+5	0.4341	0.4659
s6	50	5.7842E+6	*5.4887E+6	0.6249	0.5768
s7	60	7.4545E+6	*6.7478E+6	0.9276	0.7516
s8	64	9.6731E+6	*7.2236E+6	1.1276	0.8661
s9	85	2.0829E+8	*1.1196E+7	12.3077	1.6086
s10	100	2.6531E+8	*1.9765E+7	15.7125	2.0007

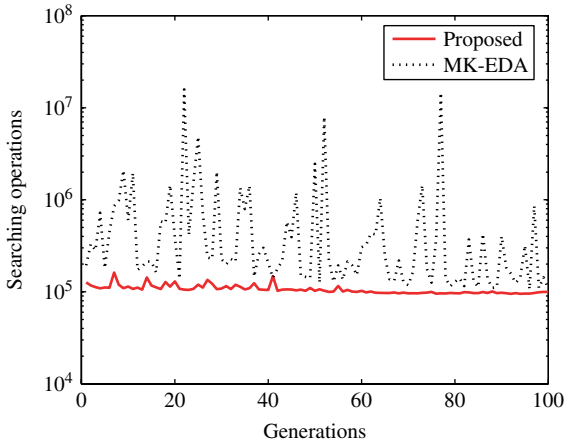

Fig. 7. The number of backtracking searching operations for instance $s9$

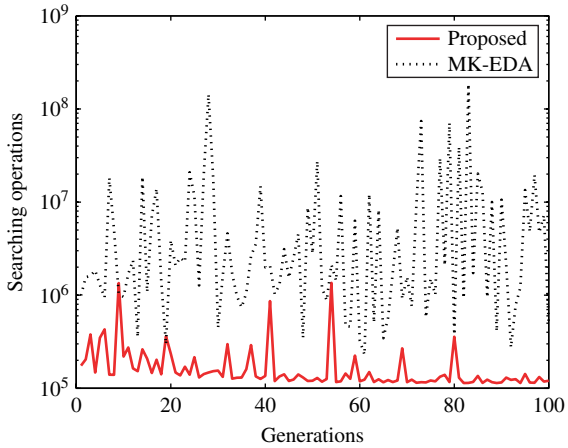

Fig. 8. The number of backtracking searching operations for instance $s10$

6. Conclusions and Further Work

In this paper, we present a novel hybrid EDA method to solve the HP model problem. For the basic k-order Markov EDA, it has very a low chance of finding the general optimum for those long-sequence and deceptive protein instances. A composite fitness function containing information of the folding structure core is introduced to replace the traditional fitness function of the HP model. It can help to select better individuals for the probabilistic model of the EDA. In addition, local search with guided operators is utilized to refine the found solutions for improving the efficiency of EDA.

The heavy computational cost is the disadvantage of the traditional backtracking method which is used to repair the invalid individuals in population. It will waste a lot of computational time for invalid closed areas of the folding structure. An improved method is proposed to reduce the computational cost of repairing for long protein sequences. A detection procedure for feasibility is added to avoid entering invalid closed areas when selecting directions for the residues. Thus, it can significantly reduce the number of backtracking searching operation and the computational cost for long-sequence proteins. It is to be noted that the improved backtracking repairing method can be used in all EA-based PSP methods that need to repair invalid individuals. And the underlying mutations are implemented for individuals in the repairing procedure.

Experimental results demonstrate that the proposed method outperforms the basic EDA method. At the same time, the proposed method is very competitive with other existing algorithms for PSP on lattice HP models. Further research is needed to determine more efficient local search strategies and probabilistic models of EDA for the protein HP model problem.

References

(1) Lau KF, Dill KA. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Marcromolecules* 1989; **22**:3986–3997.

(2) Crescenzi P, Goldman D, Papadimitriou C, Piccolboni A, Yannakakis M. On the complexity of protein folding. *Journal of Computational Biology* 1998; **5(3)**:423–466.

(3) Berger B, Leight T. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *Journal of Computational Biology* 1998; **5(1)**:27–40.

(4) Hart WE, Istrail SC. Fast protein folding in the hydrophobic hydrophilic model within three-eights of optimal. *Journal of Computational Biology* 1996; **3(1)**:53–96.

(5) Unger R, Moult J. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology* 1993; **231**:75–81.

(6) Greenwood GW, Shin J-M. On the evolutionary search for solutions to the protein folding problem. *Evolutionary Computation in Bioinformatics*, 2002; 115–136.

(7) Song J, Cheng J, Zheng TT, mao J. A novel genetic algorithm for HP model protein folding. *Proceedings of the 6th International Conference on Parallel Distributing, Computing and Applied Technologies (PDCAT-2005)*, Dalian, China 2005; 935–937.

(8) Cotta C. Protein structure prediction using evolutionary algorithms hybridized with backtracking. In *Artificial Neural Nets Problem Solving Methods*, vol. 2687. Springer Verlag: Berlin, Germany; 2003; 321–328.

(9) Krasnogor N. Self-generating metaheuristics in bioinformatics: the protein structure comparison case. *Genetic Programming and Evolvable Machines* 2004; **5(2)**:181–201.

(10) Duarte-Flores S, Smith JE. Study of fitness landscapes for the HP model of protein structure prediction. *Proceedings of CEC-2003*, Canberra, Australia, 2003; 2338–2345.

(11) Santana R, Larranaga P, Lozano JA. Protein folding in simplified models with estimation of distribution algorithms. *IEEE Transactions on Evolutionary Computation* 2008; **12(4)**:418–438.

(12) Larranaga P, Lozano JA (eds). *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers: Norwell; 2002.

(13) Lozano JA, Larrañaga P, Inza I, Bengoetxea E. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Springer-Verlag: Berlin, Germany; 2006.

(14) Mhlenbein H, Paab G. From recombination Of genes to the estimation of distributions i. binary parameters. *Proceedings of the Parallel Problem Solving from Nature-PPSN IV*, vol. 1411, *Lecture Notes in Computer Science*. Springer: Berlin, Heidelberg; 1996; 178–187.

(15) Mendiburu A, Lozano JA, Miguel-Alonso J. Parallel implementation of EDAs based on probabilistic graphical models. *IEEE Transactions on Evolutionary Computation* 2005; **9(4)**:406–423.

(16) Krasnogor N, Hart WE, Smith J, Pelta DA. Protein structure prediction with evolutionary algorithms. *Proceedings of the Genetic Evolutionary Computing Conference*, Florida, USA, 1999; 1596–1601.

(17) Hoque M, Chetty M, Dooley LS. A guided genetic algorithm for protein folding prediction using 3D hydrophobic-hydrophilic model. *Proceedings of CEC-2006*, Vancouver, BC, Canada, 2006; 2339–2346.

(18) http://www.cs.sandia.gov/tech-reports/compbio/tortilla-hp-benchmarks.html.

Benhui Chen (Non-member) received the B.E. degree in computer science from Yunnan University, China, in 1999, and received the M.E. degree in computer science from Yunnan Normal University, China, in 2005. From 1999 to 2004, he worked as a Research Associate and from 2004 to 2008 as Lecturer at Dali University, China. Since April 2008, he has been pursuing the Ph.D. degree at Waseda University, Japan. His research interests include evolutionary computation, machine learning, and bioinformatics.

Jinglu Hu (Member) received the M.Sc. degree in electronic engineering from Zhongshan University, China, in 1986, and the Ph.D. degree in computer science and system engineering from Kyushu Institute of Technology, Japan. From 1986 to 1993, he worked as a Research Associate and Lecturer at Zhongshan University. From 1997 to 2003, he worked as a Research Associate at Kyushu University. From 2003 to 2008, he worked as an Associate Professor, and since April 2008, he has been a Professor at The Graduate School of Information, Production and Systems of Waseda University, Japan. His research interests include computational intelligence such as neural networks and genetic algorithms, and their applications to system modeling and identification, bioinformatics, time series prediction, and so on. Dr Hu is a member of IEEE, IEEJ, SICE, and IEECE.