Protein Structure Prediction in 2D HP Lattice Model Using Differential Evolutionary Algorithm

Nanda Dulal Jana[1] and Jaya Sil[2]

[1] Department of Information Technology,
National Institute of Technology, Durgapur,
West Bengal, India
[2] Department of Computer Science and Technology,
Bengal Engineering & Science University, Shibpur, West Bengal, India
nanda.jana@gmail.com, js@cs.becs.ac.in

Abstract. Protein Structure Prediction (PSP) is a challenging problem in bioinformatics and computational biology research for its immense scope of application in drug design, disease prediction, name a few. Developing a suitable optimization technique for predicting the structure of proteins has been addressed in the paper, using Differential Evolutionary (DE) algorithm applied in the square 2D HP lattice model. In the work, we concentrate on handling infeasible solutions and modify control parameters like population size (NP), scale factor (F), crossover ratio (CR) and mutation strategy of the DE algorithm to improve its performance in PSP problem. The proposed method is compared with the existing methods using benchmark sequence of protein databases, showing very promising and effective performance in PSP problem.

1 Introduction

One of the greatest challenges in bioinformatics research is to solve protein folding problem, called protein structure prediction from its primary amino acids sequences. A protein is represented by a sequence of 20 different amino acids, joined end to end by formation of peptide bonds. Fig. 1 shows the peptide bond between two amino acids.

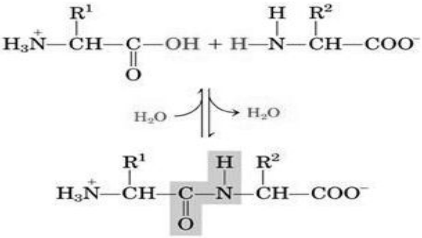


Fig. 1. Peptide Bonds among two amino acids

S.C. Satapathy et al. (Eds.): Proceedings of the InConINDIA 2012, AISC 132, pp. 281–290.
springerlink.com

The 3D structure (native structure) of a protein describes biological functions, which play an important role in drug design, disease prediction and so on. Biological scientists predict the structure of proteins by experiments like X-ray crystallography and nuclear Magnetic Resonance (NMR) [1]. However, these processes are very time consuming and expensive too, so researchers concentrate on protein structure prediction using computational strategies.

Even for a small protein sequence, exhaustive search is impossible due to the exponential growth in the number of possible conformations with the number of amino acids. Moreover, the computational analysis of the prediction of structures is intractable by using simple lattice models [2]. To overcome the limitations, researchers use a heuristic optimization method, in particular evolutionary algorithms [3, 4, 5] to predict 3D protein structure. In this work, simple 2D HP lattice model [6] has been considered where amino acids are characterized by polar (P) and non-polar (H) residues of amino acid. In this model, each H and P is embedded on 2-D square lattice with non-overlapping amino acids, called feasible conformation. In infeasible conformation, amino acids are overlapping on lattice. The total numbers of hydrophobic contacts i.e., H-H non local contacts between the amino acids, which are not adjacent in the sequence are used as energy function in this model.

In the paper, a Differential Evolutionary (DE) algorithm for protein structure prediction (PSP) problem based on the 2D HP lattice model has been presented. First infeasible conformation is converted to feasible conformations by checking possible relative movement of the amino acids. To improve the performance of the DE algorithm, selections of control parameters such as NP, F, and CR are modified. Finally, results produced by this algorithm are compared with previously published results.

The paper is structured as follows: Section 2 presents the preliminaries of 2D HP lattice model and section 3 describe the Differential Evolutionary Algorithm briefly. Methodology for applying DE algorithm to PSP problem is described in section 4. In section 5, the experimental results are compared against other known algorithms. Finally, conclusion and future direction is summarized in section 6.

2 2D HP Model

The most widely used discrete model for protein structure prediction is 2D HP lattice model [6]. In this model each amino acid is classified as hydrophobic or non-polar (H) or a hydrophilic or polar (P) based on their hydrophobicity. Conformation of a protein is then represented as a self-avoiding walk i.e., a feasible conformation in a 2D HP square lattice. The basic concept of this model is that the hydrophobic (H) amino acids lying in its core to provide more stable structure with minimum free energy. Each hydrophobic (H) amino acids tend to avoid interact with solvent environment and hence tend to move inside the structure where polar amino acids remain on the outside of the structure.

An H-H non local bond is a pair of Hs that are adjacent in the lattice but not in the sequence. The native conformation of a protein corresponds to the minimum free energy conformation for that protein. The optimal feasible conformation in the square 2D HP lattice model is one that has the maximum number of H-H non local bonds, which give minimum energy value. In Fig. 2, the black circles and the white circles represent hydrophobic (H) amino acids and hydrophilic (P) amino acids respectively. ‘S’ and ‘E’ represent the starting amino acid and end amino acid while dotted lines represent H-H non local contacts. The conformation has 9 H-H non local contacts and this is the maximum number of contacts for the given sequence.

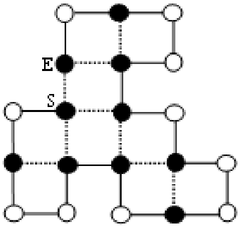


Fig. 2. An optimal conformation for the sequence HPHPPHHPHPPHPHHPPHPH in the 2D square lattice

3 Differential Evolutionary (DE) Algorithm

Differential Evolutionary (DE) algorithm [7, 8, 9] is a population based powerful stochastic search method for global optimization problem. It is applied on the optimization problem when the possible solutions are represented by a real-valued vector. In this algorithm, an individual is a set of D optimization parameters, represented by a D-dimensional parameter vector. Like other evolutionary algorithms, DE represented by an initial population (P), is a set of NP, D-dimensional parameter vectors. The generation G is denoted by G=0, 1, 2,….,G_{max}. The i^{th} vector of the population in the current generation G is given as-

$$\vec{X}_{i,G} = \{x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \ldots, x_{D,i,G}\}$$

Here $x_{j,i,G}$ is the j^{th} component (j=1 … D) of the i^{th} parameter vector (i=1 ... NP) at generation G and the value of the parameter is randomly generated using a uniform distribution $x_j^{low} \le x_{j,i,G} \le x_j^{up}$, where x_j^{low} and x_j^{up} is the lower and upper limit of $x_{j,i,G}$. After initialization of the parameter vectors, DE enters in a cycle to execute the steps: mutation, crossover and selection as described below.

3.1 Mutation

Mutation operation creates a donor vector $\vec{V}_{i,G}$ corresponding to each population or target vector $\vec{X}_{i,G}$ in the current generation G. Different mutation strategies used in several optimization problems are described in [10]:

DE/rand/1: $\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F.(\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i})$

DE/best/1: $\vec{V}_{i,G} = \vec{X}_{best,G} + F.(\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i})$

DE/best/2: $\vec{V}_{i,G} = \vec{X}_{best,G} + F.\left(\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}\right) + F.(\vec{X}_{r_3^i,G} - \vec{X}_{r_4^i})$

De/rand/2: $\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F.\left(\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}\right) + F.(\vec{X}_{r_4^i,G} - \vec{X}_{r_5^i})$

Where the indices $r_1^i, r_2^i, r_3^i, r_4^i, r_5^i$ are mutually exclusive integers, randomly chosen from the range [1, NP] and different from the base vector *i*. For each donor vector, these indices are generated randomly. Here F ∈ [0, 2] [11] is a positive control parameter, known as scaling factor used to scale the difference vectors. $\vec{x}_{best}$ is the best individual vector in the population at generation G.

3.2 Crossover

Crossover operation plays an important role to explore the search space in DE. The crossover operation is applied to each pair of the target vector $\vec{X}_{i,G}$ and its corresponding donor vector $\vec{V}_{i,G}$ to generate a trial vector $\vec{U}_{i,G} = \left\{u_{1,i,G}, u_{2,i,G}, u_{3,i,G}, \ldots\ldots, u_{D,i,G}\right\}$. In DE, there are two types of crossover techniques: binomial and exponential crossover [7]. In binomial crossover, the trial vector obtained as

$$\vec{U}_{j,i,G} = \begin{cases} v_{j,i,G}, & if\ (rand \leq CR\ or\ j = j_{rand}) \\ x_{j,i,G}, & otherwise \end{cases}$$

Where *rand* ∈ [0, 1], is a uniformly distributed random number for *j*th component of *i*th parameter vector. CR ∈ [0, 1] is a crossover probability, which defines by user that controls the parameter values are copied from the donor vector. $j_{rand} \in$ [1, D] is a randomly chosen index, which ensures that $\vec{U}_j$ gets at least one component from $\vec{V}_{i,G}$

In exponential crossover, first an integer *n* ∈ [1, D] is chosen randomly. This integer acts as a starting point in the target vector from where the crossover or exchange of components starts with the donor vector. Another integer L is chosen from the interval [1, D] where L denotes the number of components the donor vector actually contributes to the target. After choosing *n* and L, the trial vector is obtained as

$$\vec{U}_{j,i,G} = \begin{cases} v_{j,i,G}, & for\ j = \langle n\rangle_D, \langle n+1\rangle_D, \langle n+2\rangle_D, \ldots\ldots, \langle n+L-1\rangle_D \\ x_{j,i,G}, & for\ all\ other\ j \in [1, D] \end{cases}$$

Where $\langle\rangle_D$ denote modulo function with modulus D. The integer L∈ [1, D] is taken according to the following pseudo-code:

```
L=0; Do
  {   L=L+1; } While ((rand (0, 1) < CR) and (L < D))
```

Where CR is the crossover probability. Hence in effect, probability (L ≥ *v*) = $(CR)^{v-1}$ for any *v* >0.

3.3 Selection

The next stage of DE algorithm is selection operation to determine whether the target vector or the trial vector survives to the next generations i.e., at G=G+1. The selection operation is describe as

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & if\ f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G}) \\ \vec{X}_{i,G} & otherwise \end{cases}$$

Where $f(X)$ is the function to be minimized. Depending on the stopping criteria, the above three stages are repeated generation after generation.

4 Methodology

4.1 Vector Encoding

The performance of an evolutionary algorithm is strongly depending on the way of representing the individuals, a set of optimization parameters. To date, there are three ways to represent a conformation of a protein on a lattice [6]: Distance matrix, Cartesian coordinates and internal coordinates. In this work, internal coordinates are used, which are two types [12]: Absolute internal coordinate and Relative internal coordinate. In absolute internal coordinate, according to the axis of the lattice, a given amino acid moves. The conformation, using this scheme are coded with a sequence in $\{N, S, E, W\}^{n-1}$, which corresponds to North, South, East and West for n length protein sequence in 2D lattice. But, in Relative internal coordinate encoding, a given amino acid moves according to the previous amino acid movements. Using this encoding scheme, conformation is represented with a sequence in $\{F, L, R\}^{n-1}$, which corresponds to Forward, Left and Right for n length protein sequence in the plain.

In DE, every individual are real valued vectors, decoded into a specific conformation of a protein on the 2D square lattice. Therefore, an adaptation concept is necessary for encoding and decoding the sequence of movements of a protein on the lattice. The same adaption concept proposed in [5] has been used in the paper. Using relative internal coordinate in 2D square HP lattice model, the movements are Forward, Left and Right. Therefore, the phenotypical representation of a solution is defined over the alphabets {F, L, R}. The genotypical representation is still a real valued vector. Consider $x_{i,j}$ is the j^{th} element of individual and P is the string representing the sequence of movements of the conformation and $\alpha < \beta < \gamma < \delta$ arbitrary constants in $\mathbb{R}$. The genotype-phenotype mapping is defined as follows:

$$\begin{aligned} &if\ \alpha < x_{i,j} \leq \beta \quad then\ P_j = L \\ &if\ \beta < x_{i,j} < \gamma \quad then\ P_j = F \\ &if\ \gamma \leq x_{i,j} < \delta \quad then\ P_j = R \end{aligned}$$

In this work, $\alpha = -3$, $\beta = -1$, $\gamma = 1$, $\delta = 3$ are considered as in [5].

4.2 Initial Population

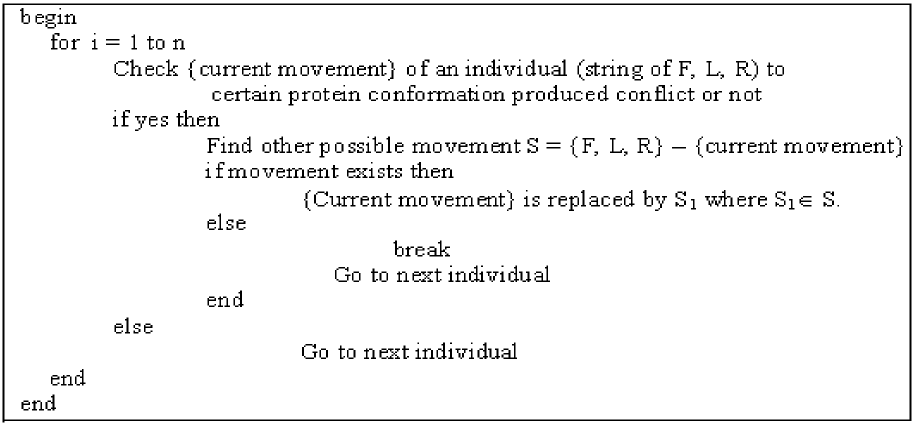
Initial population has been generated randomly using the relative internal coordinate encoding scheme. Therefore, the conformation of a protein is represented by a string of alphabets F, L, and R. Using this scheme, the conformations of proteins may be feasible (non overlapping of amino acids on the lattice) or infeasible (overlapping of amino acids). Thus, using the above defined genotype-phenotype mapping we convert the string of conformation to real valued vector, because in Differential Evolutionary algorithm every individual is real valued vector. When evaluating the fitness (maximum number of H-H non local bonds) of a conformation, again the individual (real valued vector) is converted to string of alphabets F, L, and R using the same genotype-phenotype mapping. We assume, the infeasible conformations are given a fitness of -1 and a mechanism is proposed to convert the infeasible conformation to feasible one.

4.3 Proposed Mechanism

Basically, there is no fixed technique for converting infeasible conformation to feasible ones. In this mechanism, an infeasible protein conformation (string of characters F, L, R) is taken as inputs. First, we check a movement of the string one by one from the starting movement to end of infeasible conformation to check whether conflict (existences of overlapping) is occurred or not. If any conflict occurred with the movement, then we check the possible movement except the current movement resulting nonoccurrence of conflict. If one possible movement exists, we replace the current movement by finding new movement and rest of the movements is unchanged in which no conflict occurs. If there is two possible movements exist, select any one arbitrarily. This checking procedure is repeated through the rest of the movements in the infeasible conformation. If there is no possible movement, we consider this conformation is an infeasible conformation and assign the fitness to -1. Since, our objective is to maximize the number of H-H non local bonds using the DE therefore, after some generation infeasible conformation has been removed from the population. The proposed algorithm is shown in Fig. 3. In Fig. 4, (a) is an infeasible conformation with string 'FFLLRLL**F**LR'. The movement F creates a conflict with 1st and 9th amino acids. There are two movements: L and R are to be checked. But L movement also creates a conflict with 5th amino acid. Therefore, only R movement is possible as shown in (b) where the feasible conformation is FFLLRLL**R**LR. In Fig. 4, (c) is the infeasible conformation where (d) and (e) are the two possible feasible conformations. In Fig. 4, (f) is an infeasible conformation with string 'FLFLFLL**F**F'. The movement F creates a conflict with 3rd and 9th amino acids. Two movements L and R conflict with 5th and 1st amino acid and so this remains as infeasible conformation.

4.4 DE control Parameters

The mutation strategy, crossover strategy and control parameters such as the population size (NP), crossover ratio (CR) and the scale factor (F) are strongly influence the performance [10, 13] of the DE algorithm. Therefore, it is necessary for appropriate combination of strategy and their associated parameter values to solve specific optimization problems. In DE, larger population size explores the search space but decrease the probability to find the correct search direction. In this work, we used the population size (NP) from 5D to 10D (D is the dimension of the problem) [8].



```
begin
  for i = 1 to n
        Check {current movement} of an individual (string of F, L, R) to
              certain protein conformation produced conflict or not
        if yes then
              Find other possible movement S = {F, L, R} – {current movement}
              if movement exists then
                    {Current movement} is replaced by $S_1$ where $S_1 \in$ S.
              else
                          break
                    Go to next individual
              end
        else
                    Go to next individual
  end
end
```

Fig. 3. Algorithm for converted infeasible conformation to feasible conformation

The exploration and exploitation of the DE algorithms is very sensitive to the selection of mutation strategy. The donor vectors are created using mutation strategy. The most widely used strategy are DE/rand/1/- and DE/best/1/-. The first strategy is responsible for exploring the search space and the other is used for fast convergence to global optima. Initially, we used DE/best/1/- strategy but if no improvement in best fitness have been seen with N number of generations, then change to strategy DE/rand/1/- up to M number of generations. If fitness is improved within M generations, back to the initial strategy, otherwise back to the initial strategy after M generations. Here, we also consider one difference vector to be perturbed because more difference vectors increase the convergence speed at the cost of possibility to trap at local optima. The scale factor (F) has great importance to the DE algorithm. The large values of F are used for escaping the solution from a local optimum and small values provide rapid convergence but high probability to trap to local optima. Therefore, we used F value from 0.5 to 0.9 at each generation to generate the donor vector. If some components of the donor vector violate its limits, then set the corresponding component to a random value within the specified limits of that component.

In this paper, exponential crossover with crossover probability (CR) from 0.8 to 1. Since, large crossover rate speed up the convergence [11]. Here objective is to find the maximum number of H-H non local contacts.

5 Results and Discussion

In this section, we explain the results obtained by the improved DE algorithm on various benchmark sequence [14] and compare them with the results of protein structure prediction by Genetic algorithm [14], Multimeme Algorithm [15], DE approach [5] and hybrid DE [16]. We are considering 50 runs for each benchmark sequence using different random seeds. For the experiments, we used the following parameters: NP ∈ [5D, 10D], F ∈ [0.5, 1] and CR ∈ [0.8, 1]. To explore the search space, alternatively use the strategy DE/best/1/exp. and DE/rand/1/exp. Using the above strategy adaption, we consider N=50 and M=70. The algorithm was developed in MatLab 2010b and run on a PC 2.26 GHz core 2 duo with 2 GB RAM under Windows XP.

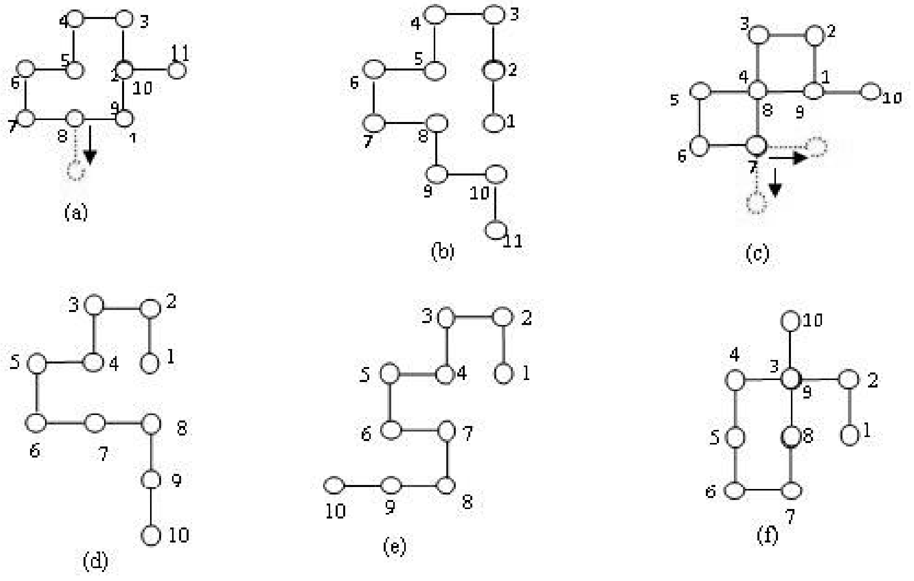


Fig. 4. (a) infeasible conformation of 11 lengths protein sequence; (b) feasible conformation of (a); (c) infeasible conformation with two possible movements; (d) and (e) are the two feasible conformations of (c) and (f) infeasible conformation

The benchmark sequences are shown in Table 1. These sequences of proteins are not the real world proteins but benchmark for 2D HP square lattice model. In Table 1, H^i, P^i and $(HP)^i$ represents the repetitions of the respective amino acids while C_{max} represents the maximum number of H-H non local contacts known to date.

Table 2 shows the results of the proposed approach and other evolutionary algorithmic approaches. In this table, 1st, 2nd and 3rd column shows the sequence number, length of the sequence and maximum (C_{max}) H-H non local contacts respectively. The 4th, 5th, 6th and 7th column represent C_{max} using Genetic Algorithms [14], Multimem Algorithms (MMA) [15], Differential Evolution approach [5] and hydrid DE [16]. Blank space in 5th column represents that the corresponding sequence is not considered. Last column, split by two: first, maximum H-H contacts are obtained and the number of times this maximum was found within the parenthesis in 50 independent runs. Next, the average number over 50 independent runs is listed in the last column.

Result using the proposed approach is better or equal than the GA technique for all the sequences. For the sequence S1, S3, S4, S6 and S8 have equal C_{max} in both MMA and the proposed one. For S5, we obtained better result over MMA while C_{max} are same with the results by hybrid DE and DE approach except for the sequence S8.

Table 1. Benchmark sequence for 2D HP square lattice

Seq.No.	**HP Chain**	**Length**	**C_{max}**
S1	$HPHP^2H^2PHP^2HPH^2P^2HPH$	20	9
S2	$H^2P^2HP^2HP^2HP^2HP^2HP^2HP^2H^2$	24	9
S3	$P^2HP^2H^2P^4H^2P^4H^2P^4H^2$	25	8
S4	$P^3H^2P^2H^2P^5H^7P^2H^2P^4H^2P^2HP^2$	36	14
S5	$P^2HP^2H^2P^2H^2P^5H^{10}P^6H^2P^2H^2P^2HP^2H^5$	48	23
S6	$H^2PHPHPHPH^4PHP^3HP^3HP^4HP^3HP^3HPH^4PHPHPHPH^2$	50	21
S7	$P^2H^3PH^8P^3H^{10}PHP^3H^{12}P^4H^6PH^2PHP$	60	36
S8	$H^{12}PHPHP^2H^2P^2H^2P^2HP^2H^2P^2H^2P^2HP^2H^2P^2H^2P^2HPHPH^{12}$	64	42

Table 2. Comparison of Results using different approaches

Seq. No.	Length	C_{max}	GA[14]	MMA [15]	Hybrid DE[16]	DE[5]	Our Approach	
							Max	Average
S1	20	9	9	9	9	9	9(50)	9.00
S2	24	9	9		9	9	9(50)	9.00
S3	25	8	8	8	8	8	8(50)	8.00
S4	36	14	14	14	14	14	14(50)	14.00
S5	48	23	22	22	23	23	23(45)	22.88
S6	50	21	21	21	21	21	21(50)	21.00
S7	60	36	34		35	35	35(42)	34.82
S8	64	42	37	39	42	42	39(40)	38.80

In this work, we considered smaller population size (NP), random scale factor (F) and random crossover rate (CR) within the defined range. These are the different from hybrid DE and DE approach in which they considered large population size and fixed F and CR value. Also, we proposed a mechanism which is different from the repair process in hybrid DE that converts the infeasible conformations to feasible conformations. consequently, our algorithms took few seconds to complete one run up to the 50 length sequence and for 60 and 64 took average time 150 to 1000 seconds per run.

6 Conclusion and Future Work

In this paper, we proposed an improved DE algorithm for protein structure prediction using the 2D HP square lattice model. Our algorithm combines with the mechanism that converts infeasible conformation to feasible conformation. Random values of Scale Factor (F) and Crossover Ratio (CR) within the specific limits improves the performance of DE algorithm. Selection of small population size (NP) gives faster run within a specific generation. Experimental results on the benchmark sequences show that the proposed approach is promising and effective than GA and MMA and also from standard DE approach with respect to NP and number of generations. We would like to improve the performance of DE algorithm using Neighborhood Search concepts for large sequence length of proteins and like to use the DE to predict the structure of a protein on the triangular lattice model.

References

1. Unger, R.: The genetic algorithm approach to protein structure prediction. Structure and Bonding 110, 153–175 (2004)
2. Berger, B., Leight, T.: Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. Journal of Computational Biology 5(1), 27–40 (1998)
3. Unger, R., Moult, J.: A Genetic Algorithm for Three Dimensional Protein Folding Simulations. In: Proceedings of the 5th Annual International Conference on Genetic Algorithms, pp. 581–588 (1993)
4. Pedersen, J.T., Moult, J.: Protein Folding Simulations with Genetic Algorithms and a Detailed Molecular Description. J. Mol. Biol. 269, 240–259 (1997)
5. Bitello, R., Lopes, H.S.: A differential evolution approach for protein folding. In: Proc. IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 1–5 (2006)
6. Dill, K.A.: Theory for the folding and stability of globular proteins. Biochemistry 24, 1501 (1985)
7. Storn, R.M., Price, K.V.: Differential Evolution- a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, International Computer Science Institute, Berkeley, USA (1995)
8. Storn, R.M., Price, K.V.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
9. Storn, R.M., Price, K.V., Lampinen, J.A.: Differential Evolution – A Practical Approach to Global Optimization. Springer, Berlin (2005)
10. Das, S., Suganthan, P.N.: Differential Evolution: A survey of the state-of –the-art. IEEE Transaction on Evolutionary Computation 15(1) (2011)
11. Storn, R.: On the usage of differential evolution for function optimization. In: Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS), pp. 519–524. IEEE, Berkeley (1996)
12. Krasnogor, N., Hart, W.E., Smith, J., Pelta, D.A.: Protein structure prediction with evolutionary algorithms. In: Proc. Int. Genetic and Evolutionary Computation Conf., pp. 1596–1601 (1999)
13. Liu, J., Lampinen, J.: On setting the control parameter of the differential method. In: Pro. 8th Int., Conf. Soft Computing (MENDEL 2002), pp. 11–18 (2002)
14. Unger, R., Moult, J.: Genetic Algorithms for protein folding simulations. Journal of Molecular Biology 231(1), 75–81 (1993)
15. Krasnogor, N., Blackburne, B.P., Burke, E.K., Hirst, J.D.: Multimeme Algorithms for Protein Structure Prediction. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 769–778. Springer, Heidelberg (2002)
16. Santos, J., Diéguez, M.: Differential Evolution for Protein Structure Prediction Using the HP Model. In: Ferrández, J.M., Álvarez Sánchez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2011, Part I. LNCS, vol. 6686, pp. 323–333. Springer, Heidelberg (2011)