A Hybrid Genetic Algorithm for 2D Protein Folding Simulations

Hamza Turabieh
Information Technology Department,
CIT Collage
Taif University, KSA

ABSTRACT
Protein folding problem is one of the most interesting problem in the medical field, which consists in finding the tertiary structure for a given amino acid sequence of a protein. Protein folding is NP hard problem. In this paper, we hybridized genetic algorithm with a local search algorithm to solve 2D Protein folding problem. This kind of hybridization empower the genetic algorithm exploration and exploitation process. The local search algorithm used is great deluge algorithm, which focus on intensification process. The experiments conducted in this work have shown the good performance of the proposed algorithm compared to similar approaches of the state of the art when dealing with different protein folding optimization problems. In particular, a good tradeoff between search space diversication and intensication is achieved. Possible extensions upon this hybridization are also discussed.

Keywords
Protein folding, Genetic algorithm, Great deluge, 2D HP Model.

1. INTRODUCTION
Recently, prediction of a proteins structure from its amino-acid sequence is one of the most exciting problems in molecular biology, computational biology, biochemistry and physics. Proteins functions are quite diverse, for example myosin and actine are involved in muscle contraction, hemoglobin is responsible for the oxygen transport in the blood, structural proteins determine the structure of cells, other proteins help in the control of brain signals, and so forth.

This problem has been widely studied under the HP model in which each amino acid is classified, based on its hydrophobicity, as H (hydrophobic or non-polar) or P (hydrophilic or polar). The amino acid sequence comes in two and three-dimensional shape of a protein. Any protein can spontaneously fold into a stable unique native conformation, which is influenced by cellular environment surrounding the polypeptide chain. Once the proteins complete the folding process, it will be active and are in their native state. As a result, the protein function depends on its tertiary structure, which is depends on the amino acid sequence. Whilst, any inaccurate folding will leads to a loss of the protein function, which can cause several sporadic and genetic diseases such as Alzheimers and Parkinson [1]. Understanding of protein folding process would definitely lead to an improved treatment of these diseases.

Lau and Dill [2] proposed the hydrophobic-hydrophilic model, which is a free energy model. The free energy of native conformation of a protein can be evaluated based on the relation between hydrophobic amino acids. The amino acid sequence of a protein is modeled as a binary sequence of hydrophobic and hydrophilic amino acids. Some amino acids cannot be classified clearly as being either hydrophobic or hydrophilic, however, this model ignore this facts to propose a simple model. This model commonly referred to HP model, where H presents for hydrophobic and P for polar.

Developing algorithms for solving protein folding structure are supportive tools for contemporary molecular biology. Moreover, recent computational analysis improve that this problem is intractable on simple lattice model [3]. Heuristic and meta-heuristic optimization algorithms seem the most suitable choice to solve protein folding optimization problem. As a results, many researchers applied heuristic and meta-heuristic algorithms to find a stable native state for different protein size. Unger and Moult applied genetic algorithm to solve protein folding problem [4,5], which is consider an early application of genetic algorithms. They used only feasible solutions and applied crossover only when the chain is rejoint at 0, 90 and 270 degree angle. They involve coordinates that specify an absolute direction on square and cube lattice. Patton et al. [6] applied genetic algorithm with coordinate representation based on relative direction. They outperform the genetic algorithm used by Unger and Moult [4].

Krasnogor et al. [7] investigate what is the appropriate mix of evolutionary operators such as (crossover, mutation and micromutation) and its probability to solve protein folding problem. Based on their experimental results, they found that single point crossover is not able to transfer building blocks and micromutation works as a local search mechanism. They conclude that a small probability of crossover and high mutation and micromutaion probability is the best combination for genetic algorithm parameters. Jiang et al [8] applied a hybrid algorithm by combining genetic algorithm with tabu search; the tabu is used to perform the crossover operation. The authors compare their results with standard genetic algorithm, their proposed method outperform standard genetic algorithm.

Liang and Wong presented an Evolutionary Monte Carlo Algorithm, which incorporates the genetic algorithm and simulated tempering [9]. Ramakrishnan et al. proposed dynamic Monte Carlo dynamic Monte Carlo for solving protein folding problem [10]. Shmygelska and Hoos applied Ant Colony Optimization Algorithm for the 2D HP model [11] and in [12] an improved version for both 2D and 3D. Interested readers can find more details about protein folding structure research in the comprehensive survey paper by [13,14].

The remainder of this paper is structured as follow. Section 2 describes 2D HP model for protein folding. Section 3 illustrates Genetic Algorithm, great deluge algorithms and the hybrid algorithms used to solve 2D HP model. The results and findings are presented in Section 5. Finally, conclusion remarks are made in Section 6.

2. 2D HP LATTICE MODEL

One of the famous considered protein models is the hydrophobic-hydrophilic model (HP model), which was proposed by Lau and Dill [2]. HP model abstract the hydrophobic interaction process on protein folding by reducing a protein to a heteroplymer that represents a predetermined pattern of hydrophobicity in the protein. The amino acids (*n*) are classified as hydrophobic amino (H) or polar (P). The feasible protein folds are represented as a self-avoiding path on a lattice in which vertices represented by the amino acids (either H or P). The energy potential of the HP model reflects the fact that hydrophobic acids have the propensity to form a hydrophic core. To achieve this feature of protein folding, the protein folding adds a value ϵ (Typically $\epsilon = -1$) for every pair of hydrophobics that form by a pair of amino acids that are adjacent on the lattice and not consecutive in the sequence. For example, Figure1 presents a particular conformation of a protein where the black beads present hydrophobic and white beads present polar amino acids. The dotted lines represent the H-H contacts. The fold in Figure1 has energy of -4. Formally, a native state is a conformation having minimum contact energy [2].

Formally, a native state is a conformation having minimum contact energy as following:

$$Min \;\; E = \sum_{1 \le i+1 \le j \le n} B_{i,j}\,\delta(r_i, r_j)$$

where:

$$\delta(r_i, r_j) = \begin{cases} 1 & if \;\; \|r_i - r_j\| = 1 \; and \;\; i \neq j \pm 1 \\ 0 & otherwise \end{cases}$$

and :

$$B_{i,j} = \begin{cases} 1 & if \;\; i \;\; and \;\; j \;\; are \; hydrophobic \\ 0 & otherwise \end{cases}$$

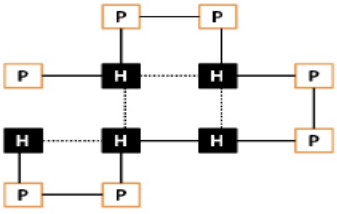


Fig 1: . A protein configuration for the sequence S = PHPPHPPHHPPH, of length 12. Black beads present hydrophobic amino acids, while white beads present hydrophilic ones. The value of the energy function for this configuration is -4.

3. THE ALGORITHM

In this section, the Genetic algorithm have been presented first, then the construction algorithm for protein folding problem based on Semi-greedy algorithm. The local search algorithm (Great deluge) is illustrated. Finally our hybrid genetic algorithm proposal is detailed

3.1 Genetic Algorithm

One of the most well known evolutionary algorithm is a Genetic Algorithm. Which is developed based on the Darwinian evolution theory [15]. It is used to search large, nonlinear solution space where expert knowledge is lacking or difficult to encode. Moreover it requires no gradient information, evolves from one population to another and produces multiple optima rather than single local one. These characteristics make GA a well-suited tool for 2D HP protein folding problem.

Genetic algorithm codes the candidate solutions of an optimization algorithm as a string of characters which are usually binary digits. In accordance with the terminology that is borrowed from the field of genetics, this bit string is usually called a chromosome (i.e. individuals). A number of chromosomes generate what is called a population.

The evolutionary process of GAs starts by the computation of the fitness of the each individual in the initial population. While stopping criterion is not yet reached we do the following: Select individual for reproduction using some selection mechanisms (i.e. tournament, rank, etc). Then create an offspring using crossover and mutation operators. The probability of crossover and mutation is selected based on the application that will be solved. Compute the new generation of GAs. This process will end either when the optimal solution is found or the maximum number of generations is reached. A flowchart for GA process is presented in Figure 2.

Selection is the process which guides the evolutionary algorithm to the optimal solution by preferring chromosomes with high fitness. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated, using some measure of fitness. To create the next generation, new chromosomes, called offspring, are formulated by using some operators called crossover and mutation. Thus, a new generation will be created by selecting the best chromosomes (parents) from the previous generation and the best chromosomes from the offspring.

After several generations of creation the algorithm hopefully converges to the optimal solution or at least the optimal domain of solution. After computing the fitness of each individual, a new population must be created. To do this, two operators borrowed from natural genetic, crossover and mutations, are used. Crossover operator is used to produce new pairs of individuals from their parents. The produced individuals (i.e. childes) have many features from their parents. There is a high probability that the child's will provide a better fit to the problem.

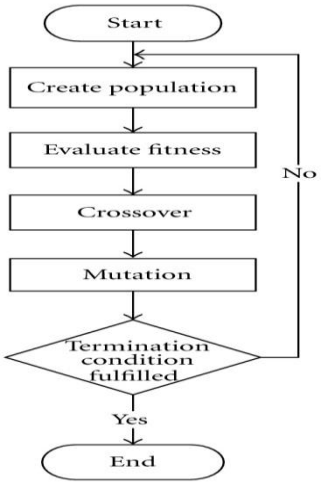


Fig 2. Flowchart of a simple GAs process

3.2 Great Deluge Algorithm

The Great Deluge algorithm (GD) is a meta-heuristic algorithm applied to solve hard optimization problems, which was introduced by Dueck (1993) [16]. It is a local search procedure which has certain similarities with simulated annealing but has been introduced as an alternative. This approach is far less dependent upon parameters than simulated annealing. It needs just two parameters: the amount of computational time that the user wishes to "spend" and an estimate of the quality of solution that a user requires. Apart from accepting a move that improves the solution quality, the great deluge algorithm also accepts a worse solution if the quality of the solution is less than or equal the level. In this work, the "level" is initially taken from Genetic algorithm. The GD terminate when the solution reach the estimated quality. The search continues until the bound reaches the lower limit (estimated quality). The pseudo code for our implementation of the great deluge algorithm is presented in Figure 3.

```
Set initial solution as Sol_best taken from GA
Calculate the initial cost function value, f(Sol);
Set best solution, Sol_best ← Sol;
Set estimated quality of final solution, estimatedquality from the user.
Set number of iterations, NumOfIte;
Set initial level: level ← f(Sol);
Set decreasing rate
   β = ((f(Sol)–estimatedquality)/(NumOfIte);
Set iteration ← 0;
Set not_improving_counter ← 0;
do while (iteration < NumOfIte)
Define neighbourhood of Sol by randomly assigning HP to a valid
position to generate a new solution called Sol*;
    Calculate f(Sol*);
    if (f(Sol*) < f(Sol_best))
      Sol ← Sol*;
      Sol_best ← Sol*;
      not_improving_counter ← 0;
     else
       if (f(Sol*)≤ level)
         Sol ← Sol*;
         not_improving_counter ← 0;
       else
          Increase not_improving_counter by 1;
        if (not_improving_counter ==
          not_improving_ length_GDA)
         exit;
level = level - β;
Increase iteration by 1;
end do;
```

Fig3: The pseudo code for the great deluge algorithm

3.3 Hybrid Genetic Algorithm

This algorithm is a combination of two algorithms: (1) Genetic algorithm and (2) Great deluge algorithm. The hybridization is done with an aim to combine the strength of these two algorithms in tackling the protein folding optimization problem outlined in the Section 2. The great deluge algorithm is used as a local search for genetic algorithm. The generic pseudo-code for the algorithm is shown in Figure 4.

```
Set the Genetic algorithm paramters.
Set great deluge algorithm parameters.
Generate Genetic algorithm population.
While(termination condition is not met)
   select a parents for offspring production
   recombine the "genes" of selected parents
   perturb the mated population stochastically
   Apply local search (Great deluge) algorithm
   evaluate it's new fitness
end
```

Fig4: The pseudo code for hybrid algorithm

4. SIMULATION RESULTS

The proposed algorithm was programmed using C# and simulations were performed on the Intel Pentium 4 2.33 GHz computer and tested on a standard benchmark 2D HP protein folding problem as shown in Table 1. The parameters for the GA algorithm and great deluge are chosen after some preliminary experiments is presented in Table 2.

Table 1. 2D HP Bnechmark dataset

Dataset	Length	Sequence
1	20	HPHPPHHPHPPHPHHPPHPH
2	24	HHPPHPPHPPHPPHPPHPPHPPHH
3	25	PPHPPHHPPPPHHPPPPHHPPPPHH
4	36	PPPHHPPHHPPPPPHHHHHHHHPPHHPPPPHHPPHPP
5	48	PPHPPHHPPHHPPPPPHHHHHHHHHHPPPPPPHHPPHHPPHPPHHHHH
6	50	HHPHPHPHPHHHHPHPPPHPPPHPPPPHPPPHPPPHPHHHHPHPHPH
7	60	HHPPHHHPHHHHHHHHHPPPHHHHHHHHHHPHPPPHHHHHHHHHHHPP PPHHHHHPHHPH
8	64	HHHHHHHHHHHHPHPHPPHHPPHHPPHPPHHPPHHPPHPPHHPPHH PHPHPHHHHHHHHHHH
9	85	HHHHPPPPHHHHHHHHHHHHPPPPPPHHHHHHHHHHHHPPPHHHHH HHHHHHPPPHHHHHHHHHHHHPPPHPPHHPPHPH

Table 2. Parameters setting for Genetic Algorithm

Algorithm	Parameter	Value
Genetic Algorithm	Generation Number	100000
	Population size	50
	Crossover Rate	0.6
	Mutation Rate	0.06
	Selection Method	Roulette Wheel selection
	Crossover Type	Single point
Great Deluge	Number of iterations	10000
	Estimated solution	-100

The best results, average and standard deviation out of 11 runs are shown in Table 3 with different random seed. We can see that our approach is able to enhance the initial solutions and obtain a good results. Moreover, the proposed algorithm is roubust since obtained average is close enough to the best results.

Table 3. Results of our proposed hybrid algorithm

Dataset	Length	Initial solution	Best	Average	Std. Dev.
1	20	-3	-9	-8.416	0.7930
2	24	-2	-9	-7.583	1.3790
3	25	-1	-8	-6.250	1.6026
4	36	-5	-14	-11.500	1.7816
5	48	-4	-23	-20.426	1.2401
6	50	-5	-21	-19.000	1.2060
7	60	-6	-33	-30.916	2.0207
8	64	-4	-42	-34.250	2.3789
9	85	-9	-52	-44.333	5.5814

Figure 5 shows the box plots that illustrate the distribution of solution quality for all nine datasets. In most of the cases, there is less dispersion of the output data. We can see that there are a close gap between the best, average and worse solution qualities which demonstrates that it is robust algorithm. Figure 6 is a pictorial diagrams for our results for four datasets of the best energy conformations.

From Table 4, we can see that our approach is able to obtain a high quality results better results compared to best known results in the literature i.e. Huang C. et al. [13] applied genetic algorithm based on optimal secondary structures (GAOSS); Guo et al. [17] applied a hybrid algorithm by combining local search with elastic net algorithm (ENLS); Jiang et al. [8] proposed a combining algorithm between tabu search with genetic algorithms (GTS); Liang and Wong [9] applied Evolutionary Monte Carlo (EMC); Unger and Moult [4] applied genetic algorithms to solve protein folding problem (GA).

Table 4. Comparison between results

Dataset	Our Approach	GAOSS	ENLS	GTS	EMC	GA
1	-9	-9	-9	-9	-9	-9
2	-9	-9	-9	-9	-9	-9
3	-8	-8	-8	-8	-8	-8
4	-14	-14	-14	-14	-14	-14
5	-23	-23	-23	-23	-23	-23
6	-21	-21	-21	-21	-21	-21
7	-33	-36	-36	-35	-35	-34
8	-42	-42	-39	-39	-39	-37
9	-52	-52	---	---	-52	---

5. CONCLUSIONS AND FUTURE WORK

In this work, we present an approach that combine the genetic algorithm algorithm with great deluge algorithm to solve protein folding problem. Great deluge algorithm is used as a local search algorithm. This hybridization enhamce the searching process of finding a new solutions. Whilst, great deluge algorithm focus an exploitation process. This hybrid algorithm is simple yet effective, and produce a good result across the all the benchmark problems in comparison with other approaches studied in the literature. We also believed that with the increasing complexity of protein folding problems, the proposed approach can be easily adapted with new sequences. Additionally, research on 3D protein folding problems may be a very promising direction to be tested on using this algorithm. This is subject to our future work.

6. REFERENCES

[1] Backofen R. and Will S. 2006. A constraint-based approach to fast and exact structure prediction in three-dimensional protein models. Constraints, 11(1):530.

[2] Lau KF. and Dill KA. 1989. lattice statistical mechanics model of the conformation and sequence space of proteins. Macromolecules 22:3986-3997.

[3] Huang C. and Yang X. and He Z. 2010. Protein folding simulations of 2D HP model by the genetic algorithm based on optimal secondary structures, Computational Biology and Chemistry, 34(3), 137-142.

[4] Unger R. and Moult J. 1993. Genetic algorithms for protein folding simulations. Journal of Molecular Biology, 231:75-81.

[5] Unger R., Moult J. 1993. A genetic algorithm for three dimensional protein folding simulations. In Proc of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers; 581-588.

[6] Patton W., Punch W., and Goldman E. 1995. A standard genetic algorithm approach to native protein conformation prediction. In Proceedings of 6th International Conference on Genetic Algorithms, 574–581.

[7] Krasnogor N., Hart W.E. , Smith J. , and Pelta D.A. 1999. Protein structure prediction with evolutionary algorithms. In W. Banzhaf et al., editors, Proceedings of the GECCO'99, 1596-1601, San Mateo CA.

[8] Jiang, T.Z., Hua, Q., Cui, Shi, G.H., Ma, S.D. 2003. Protein folding simulations of the hydrophilic model by combining tabu search with genetic algorithms. J. Chem. Phys. 119 (8), 4592-4596.

[9] Liang, F.M., Wong, W.H. 2001. Evolutionary Monte Carlo for protein folding simulations. J. Chem. Phys. 115 (7), 3374-3380.

[10] Ramakrishnan R., Ramachandran B., and Pekny J. F. 1997. A dynamic Monte Carlo algorithm for exploration of dense conformational spaces in heteropolymers. Chemical Physics, 106(6):2418-2425.

[11] Shmygelska A. and Holger H. 2003. An Improved Ant Colony Optimisation Algorithm for the 2D HP Protein Folding Problem. In Springer Verlag, editor, In Proceedings of the 16th Canadian Conference on Artificial Intelligence, 400-417.

[12] Shmygelska A. and Holger H. 2005. An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. BMC Bioinformatics, 6(30).

[13] Liang F. and Hung W. W. 2001. Evolutionary Monte Carlo for protein folding simulations. Journal of Chemical Physics, 115(7).

[14] Huang C. and Yang X. and He Z. 2010. Protein folding simulations of 2D HP model by the genetic algorithm based on optimal secondary structures, Computational Biology and Chemistry, 34(3), 137-142.

[15] Goldberg, D. and Holland, J. 1988. Genetic algorithms and machine learning. *Machine Learning*, 3(2):95–99.

[16] Dueck, G. 1993. New Optimization Heuristics. The great deluge algorithm and the record-to-record travel. Journal of Computational Physics 104, 86-92.

[17] Guo Y.Z., Meng, E.M., Wang, Y. 2006. Exploration of two-dimensional hydrophobicpolar lattice model by combining local search with elastic net algorithm. J. Chem. Phys. 125, 154102-154106.

7. APPENDIX

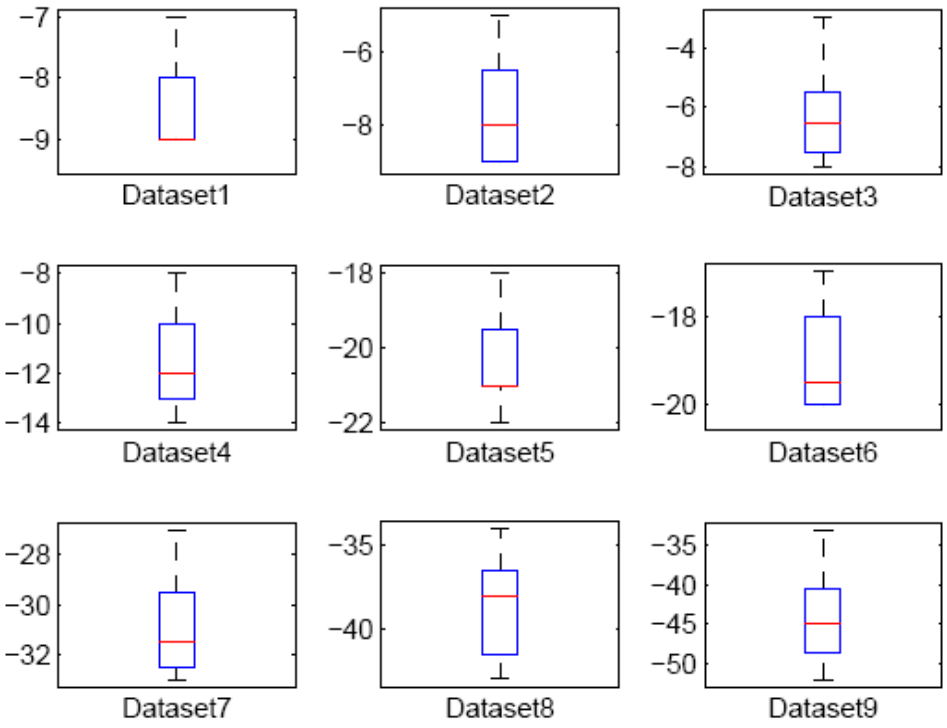
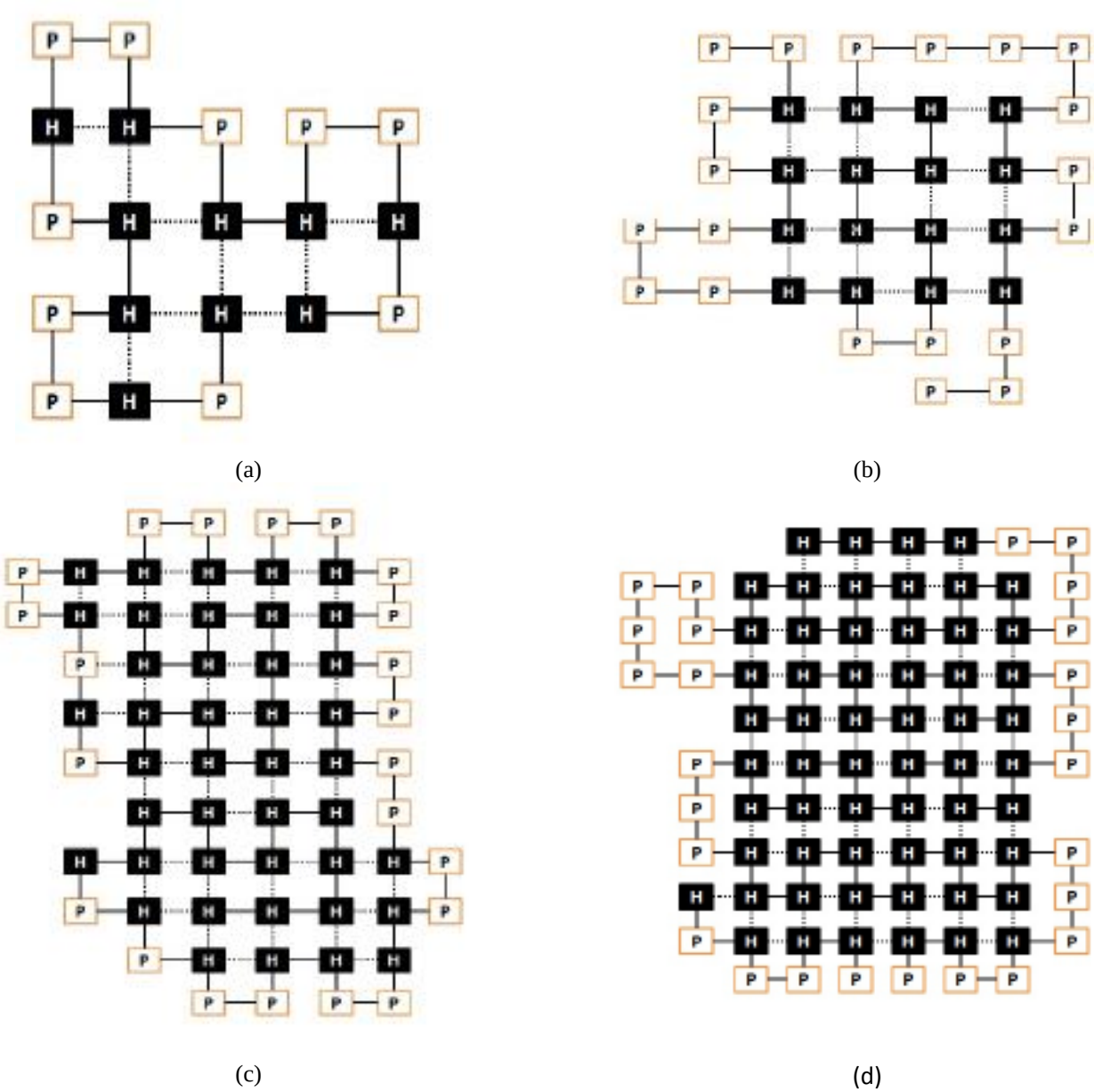


Fig5: Plot Box for protein folding



(a) (b)

(c) (d)

Fig6: Pictorial representation for best energy conformations achieved (a) Dataset 1; (b) Dataset 4; (c) Dataset 8 ; (d) Dataset 9.