Genetic algorithm for Protein Folding simulations in 2D HP model

Toño G.Quintela

October 13, 2013

Abstract

This practice is developed in the context of the subject of `Computational Intelligence` (Msc. Artificial Intelligence). The aim of this practice is to develope some genetic algorithm in order to simulate protein folding with a model of protein called HP model. Only the 2D case will be studied.

1 Introduction

1.1 Protein [1]

Proteins are large biological molecules consisting of one or more chains of amino acids. They are one of the most important elements in biology for the understanding of the metabolism of living organisms. This is due to the vast array of functionalities they perform such as catalyzing chemical reacions, replicating DNA or transporting molecules from one location to another.

One of the most interesting characteristics of proteins is that they are not operative if they aren't in their native states. In nature, mistakes during the folding process can lead to a loss of protein function, which can cause several sporadic and genetic diseases. Because of that, the direct application of a better understanding of protein folding would lead us to an improved treatment of these diseases.

1.2 Aim

In 2005, Science named the protein folding problem one of the 125 biggest unsolved problems in science [2].

Protein folding problem is a complex problem which consists of three basic subproblems.[3]

- it Folding code: The physical problem which tries to solve which are the interactions and the factors more relevant to the dynamics of the protein and tries to modelize them. In our problem we consider the HP model in 2D.
- *Protein structure prediction*: The computational problem of how to predict the native structure of a protein from its amino acid sequence.

  There are a huge number of comformations possible and the objective is to search in this conformational space for the native state. Using the model created we have to search the space under some conditions or try to optimize the energy function of the model.

  The aim of this subproblem is to find the global minimum of the energy function in the model considered.
- *Protein folding speed*: The problem of how proteins can fold so fast. Even though proteins have a vast conformational space, proteins can search and converge quickly to native states. [4] This is known as the Levinthal's 'paradox'.

  Some people think that understanding the mechanism of protein folding might lead to fast computational algorithms, for predicting native structures from their amino acid sequences.

  The aim of this subproblem is to simulate folding faithfully.

The difficulty of these problems gains the attention of scientists from diverse fields as computer scientists, physicists and chemists, in order to modelize and symplify a problem which keeps being a NP-complete problem[5].

Nowadays, it is still a multidisciplinary problem and many work teams try to tackle it.

1.3 Modeling the problem

According to the thermodynamic hypothesis the native state of a protein is that conformation which has the lowest free energy.

So, we have to define the free energy for every state and find a search algorithm to localize the native state. First of all we have to consider some simplification of the real protein, that is, a model is necessary.

The great challenge when making a model is to be able to pick up the important features while discarding some details and still being able to reproduce the major part of the phenomena.

In the protein there are different types of interactions between its components such as hydrogen bonds, van der Waals interactions and electrostatic interactions. However, the hydrophobic force is the most influential one for determining the native structure of globular proteins (proteins with a globular native conformation).[6][8]

Following thermodynamics principles we see that it is unfavourable for a protein to fold into native globular conformations because this process leads to a reduction of freedom, and thus a decreased entropy. As a consequence, there is a tradeoff between thermodynamcs and the hydrophobic force which induces a compact globular state where non-polar aminoacids are inside, minimizing their contact with the exterior (water). This is known as the hydrophobic effect which causes an increase of the entropy which compensates the decrease for adopting a globular conformation.

The assumption that the native structure of a protein corresponds to a free energy minimum is the basis for many computational simulations which attempt to predict the tertiary structure of proteins from its amino acids sequence. Consequently, the native structure is then presumed to be a conformation which minimizes the energy function over the set of possible conformations.

One of the simplest but more explanatory models used for protein folding simulations is the hydrophobic-hydrophilic (HP) model, which was introduced by Lau and Dill[7],in order to explore the whole conformational and sequence spaces of proteins. The great achievement of this model is being able to keep the main features of real protein folding despite its simplicity.

The HP model belongs to the family of lattice models. These are characterized by the following simplifications:

- uniform size of the residues (the amino acids).
- uniform bond length
- the positions are restricted to these in a regular lattice

The HP model is based on the fact that the hydrophobic force is the main force in protein folding. In this model, a protein is represented by a linear sequence of amino acids of only two types: Hydrophobic amino acids, represented by the letter H, and hydrophilic or polar amino acids, represented by the letter P.

The sequence folds into a two-dimensional (2D) square lattice. Thus, each amino acid is occupying one lattice square unit and is connected to its chain neighbours. Only self-avoiding structures are valid conformations of the sequence, that is, there can't be more than one residue in the same square unit. This sums up an extra difficulty in the problem because it is an important restriction. In fact, the self-avoidingness is a complex feature as we will see later on.

There will be two types of neighbours:

- *Connected* neighbours (adjacent along the chain sequence)
- *Toplogical* neighbours (adjacents in the lattice space and not in the chain)

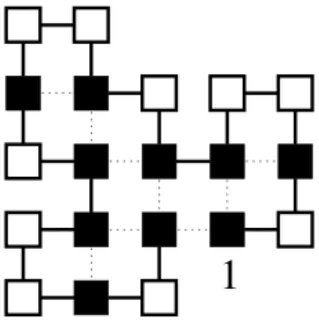

Figure 1: Protein in the HP model[9].

In order to modelize its dynamic and determine the free energy we define an energy funcion in which the only contribution to the energy is due to the H-H pairs of topological neighbors.

So, we can define the energy function of the protein by:

$$E = \sum_{1 \leq i+1 \leq j \leq n} B_{i,j}\delta(r_i, r_) \tag{1}$$

where

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

and

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

This leads to low energy conformations which are compact and have a hydrophobic core. Because of that, the hydrophobic residues tend to be inside in a low energy conformation, while polar residues are forced to the surface, since minizing this function is equal to mazimizing the number of topological H-H neighbours.

Hydrophobic cores appear in globular proteins.

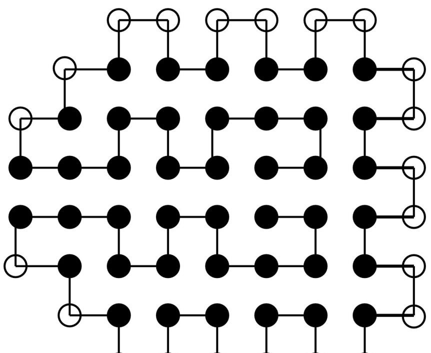

Figure 2: A core in the globular protein [10].

Our next objective is how to code the protein conformation. In order to do that in the most simplest way, we try to simplify and generalize the conformation describing it by the direction of its edges with the previous residue. The vocabulary to code them will be $\Sigma = l, r, s$ (left, right, straight). There are only $n-1$ edges, so we can define the conformations with only these variables. However, as conformations are completely rotational invariant, $n-2$ variables will be enough. This implies we do not need to represent the direction of the first edge. The conformation will be coded as a string with $n-2$ elements of the alphabet $\Sigma = l, r, s$.

To end up the coding problem, the protein's composition will be coded as a string with n elements of the alphabet: $\Sigma = H, P$

1.4 Computational problem

Once modeled, it is time to solve the computational problem. The model provides us an energy function for each conformation.

Our goal now is to search the space of conformations for the one which has the minimum energy. This problem remains a NP-complete problem. [11]

In order to solve it we have to find a randomize search algorithm. This problem has already been solved with some of them but the idea of this work is developing a genetic one.

The idea of the genetic algorithm is to mimic the natural process of evolution in biology. An initial population is selected. This is changed in different combinatorial fashions which are then recombined in order to improve the results. The conformations which give the best results are selected in some way to be the next populations.

In this genetic algorithm the search space will be all the possible conformations of the protein considered.

2 The problem nowadays

Lately the protein folding problem has become one of the problems of the decade and it seems that it will be one of the problems of the next century. The incredible velocity of the folding of proteins into the native state and the importance of the proteins theirselves in diseases make it an interesting problem.

Physicists began working with biologists in the 80's trying to modelize and simplify the problem in order to get useful works for bioengineering and pharmacology.

The incresing power of the computers allows to explore more complex and faithfully models. In the last years they've tried to use some networks systems as folding@home.

The significance of the problem make the folding@home the most power computing system over the Earth during some years[12]. It is a distributed computing project for simulating protein folding. The purposes of the project are drug design and simulations of some models. The project uses the idle processing resources of thousands of personal computers or ps3 owned by volunteers. Its primary purpose is to determine the mechanisms of protein folding, which is the process by which proteins reach their final three-dimensional structure, and to examine the causes of protein misfolding.

In the same direction was born *foldit*. It is an online puzzle video game about protein folding where players have to fold some proteins in order to get the best score possible. The results of this will be used by scientists to improve algorithms[13].

In the technical part of the protein folding problem, scientists have been trying some algorithms in order to solve this problems. The algorithms used depends on the model considered but all are parrallel-based algorithms. [14] Some of them are presented as a Constraint satisfaction problems, some others are trying to be solved with Ant colony optimization [15][16] or evolutionary algorithms as genetic ones.

3 Algorithm

3.1 Genetic algorithms

Genetic algorithms were developed by John H. Holland[17]. They have been derived from evolutionary biology, adaptation designates a process whereby a structure is progressively modified to make it better suited to its environment. A genetic algorithm is a search technique that is loosely based on simulated evolution. It uses optimization techniques inspired by biological adaptive processes such as mutation, selection and recombination. This approach makes it possible to explore a far greater range of potential solutions to a problem than conventional methods. But this generality has a tradeoff. There are other specific algorithms which exploit better the knowledge of the problem. This will perform better in this specific problems. This is the No-free-lunch theorem[18].

The generic algorithm can be described with the pseudocode showed below.

A pseducode of genetic algorithm[19]:

```
begin
  t=0;
  initialize P(t);
  evaluate procedures in P(t);
  while termination condition not satisfied do
       begin
         t = t + 1;
         select_repro C(t) from P(t-1);
         recombine and mutate structures in C(t) forming C'(t);
         evaluate structures in C'(t);
         select_replace P(t) from C'(t) and P(t-1);
       end
  end
end
```

An initial population of individual structures P(0) is generated (usually randomly) and each individual is evaluated for fitness. Then some of the individuals are selected to reproduce him (select_repro) to the mating buffer C(t). In this step better individuals are priorized to produce offspring. Recombining and mutating these we obtain C'(t). The elements of this are selected to form P(t).

3.2 Our algorithm

Population

We are codifying our possible solutions in a chain of $N-2$ digits (whence N is the number of aminoacids in the protein considered), in which every digit can take 3 possible values: $\Sigma = \{l, r, s\}$.

Not all the posible combinations of our digit values can create a valid conformation. So we will have to check the feasibility of each conformation. The biggest problem of our approach is the solution of this problem.

Initialization is one of the problems we find in the first steps. There are many different ways to initialize the populations. For this work I have considered only two:

- *Fully extended*: all the chains of the population are initialized fully extended.
- *Random Coil*: Random self-avoiding walk. It is the most faithfully representation of the problem. The protein is usually in some state similar to a random coil[20]. The problem of this is to deal with a NP complete problem of the self-avoiding walks. This is the reason why some people search solutions taking the problem as a constraint satisfaction problem. In the next figure we can see the space available considering the number of steps.

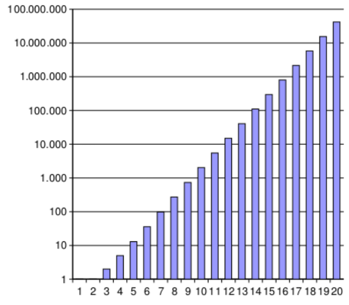

Figure 3: Space available considering the number of steps.

One of the most important factors in which the performance of the genetic algorithm improves depends on the diversity of the population. If the average distance between individuals is large, the diversity is high; if the average distance is small, the diversity is low. Getting the right amount of diversity can ensure the success of the algorithm but if the diversity is too high or too low, the genetic algorithm might not perform well.

Selection

The selection function chooses parents for the next generation using their scaled values from fitness scaling function. The function I used to check the algorithm are:

- **Stochastic uniform** lays out a line in which each parent corresponds to a section of the line of length proportional to its expectation. The algorithm make some stocastic steps moving along the line. Each step will assign the point in which it lands on to one parent.
- **Remainder**: assigns parents deterministically from the integer part of each individual's scaled value and then uses roulette selection on the remaining fractional part.
- **Uniform**: select parents at random from a uniform distribution using the expectations and number of parents. Uniform selection is not a useful search strategy, but you can use it to test the genetic algorithm.
- **Roulette**: simulates a roulette wheel with the area of each segment proportional to its expectation. The algorithm then uses a random number to select one of the sections with a probability equal to its area.
- **Tournament**: selects each parent by choosing individuals at random (I choose 4 of 20), and then choosing the best individual out of that set to be a parent.

In the results we will try to compare these methods using some examples.
In this algorithm there is elitism. The best two are selected to the next generation in order to maintain optimal or close-optimal solutions.

Mutation

Mutation functions make small random changes in the individuals in the population, which provide genetic diversity and enable the genetic algorithm to search a broader space.

Mutation developed in this practice depends on some parameter (`mutationRate`) which works as a threshold of acceptance to mute some gene. So, we have codified our protein conformation as a genome of $\Sigma = \{l, r, s\}$. We assign a random number to each gene of the chain and the ones who have bigger number than a threshold `mutationRate` their genes are randomly reassigned. This will be used in the next generation if it is a valid conformation (self-avoiding chain) and if they pass through the acceptance criteria. This criteria is a stocastic decision which favors the changes that reduce energy. It will be made with a Boltzmann factor implementing a Metropolis Montecarlo decision.

In conclusion, if the new conformation reduces the energy of his parent it will be accepted. If not we get a random number *rand* and:

$$rand < exp\left[\frac{E_i - E_f}{c}\right] \tag{2}$$

whence c is a parameter. In literature it is widely extended and it is called Temperature (because of its statistical physics origin). The attitude to reduce the value of c with the performance of the energy is called simulated annealing[21]. This technique is interesting to implement in this problem but due to my time constraints at this moment it will be keeped to future works and improvements.

Crossover

Crossover is the process to combine two individuals (parents) to form a new individual (child) for the next generation. Crossover is useful to expand the space of search of the population. There are some kinds of crossover depending on the way the parents are selected to exchange genes:

- 1-point crossover: In this process it is selected one gene. From the initial gene to the crossover point (not included) it is get from one parent and the other part from the other. In our problem the crossover point value is changed randomly in order to give flexibility and the posibility to get more range of valid conformations available. So, the gene who acts as a crosspoint don't receive gene from any of its parents.
- n-point crossover: There are selected n points over the two parents and the $n + 1$ parts are selected to the child in certain defined order.
- random crossover: Each gene of the child is selected randomly from one of his two parents.

In this practice is only implemented the first one. The others will be reserved to future works.

Not all the possible childs are valids (self-avoiding paths) so we have to introduce a control system to not allow invalid conformations get into consideration. Also, we have to reward better conformation than other, so we will use Boltzmann factors as before. Lower energy are directly selected and the others are selected in this way:

$$rand < exp\left[\frac{E_a - E_f}{c}\right] \tag{3}$$

where E_a is the average energy of the parents and c is a parameter as it is explained in the previous section.

4 Program

I used the `optimtool` of matlab for support the execution of the code I wrote. The main functions are:

- `energy_function`: It fills a square space using the indications of the conformation and searching in the neighbors of each H aminoacid to measure the energy.
- `initialization`: There are two posible implemented: Random coils and fully extended.4 In fully extended way all the conformation of all the populations are 's'. In Random coils we try to fill space aminoacid by aminoacid in order try to selecting from a random ordering posible directions. If it is not posible to continue the process is reset.
- `mutation`: The posible mutations are selected randomly. There are a threshold tunable. It call the function `acceptance` which implements the decision described before.
- `crossover`:It is only implemented the 1-point crossover. It also call to `acceptance`.

5 Results

In order to take comparisons the first thing I have do is compare between the different types of implementation.

The first comparisons will be done considering different selection processes. This comparison will be done for the same initialization method.

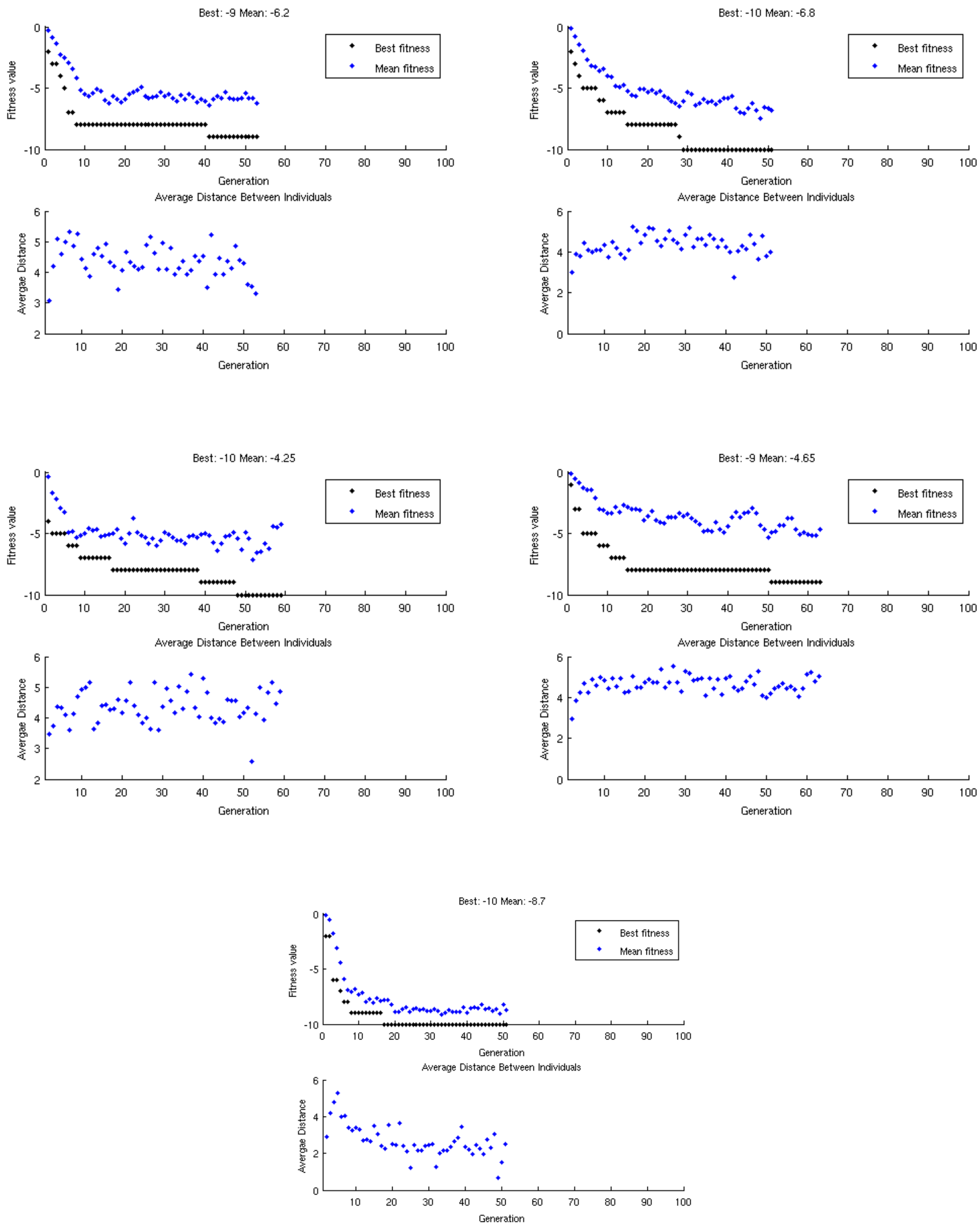

Figure 4: All of this are Fully Extended initializations for HPHHPHPHPHPHHHPHPHHHHHPHPH, from left to rigth: Stochastic uniform, remainder, roulette, uniform and tournament. We can see not only the evolution of the best and the average along generations but also the distances between them.

In this graphics is interesting to see that some of the methods don't achieve native state. Another thing interesting to see is the evolution of the distance between individuals. This graph shows a measure of diversity of population. We can see that in tournament the diversity is not only lower than others but also decreases quickly because it finds a low energy individual.

Comparing different **initializations** with a composition HHHHHHHPPHHPPHPPHHPHPPHH.
'

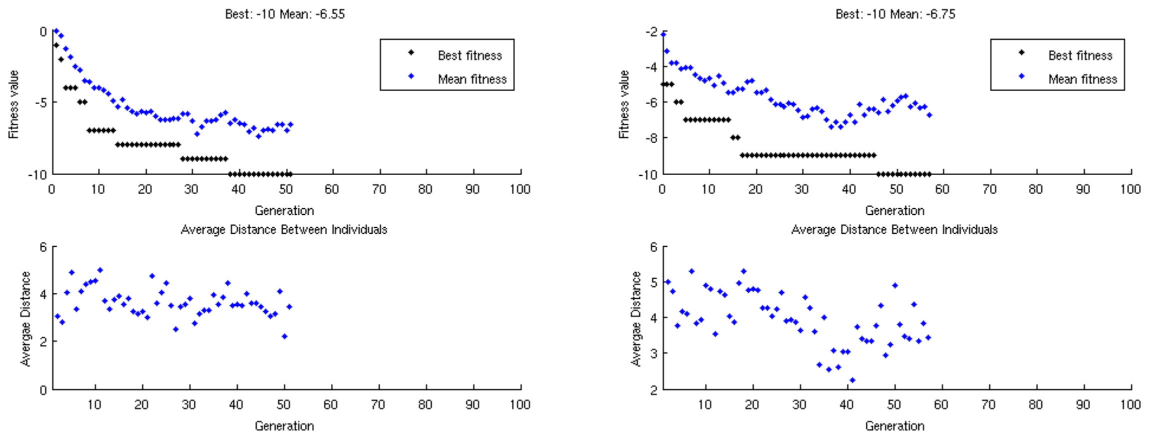

Figure 5: Remainder selection. Fully extended on the left, random coil on the rigth.

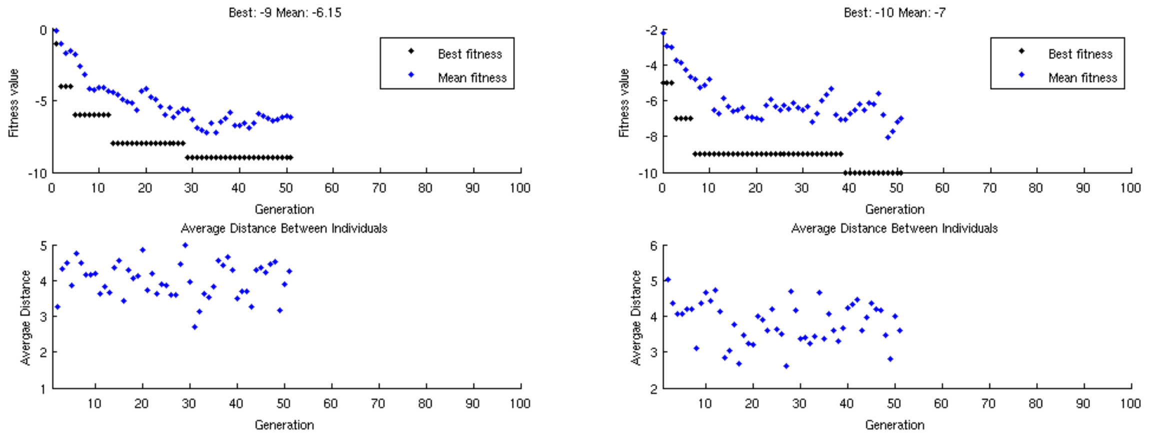

Figure 6: Roulette selection. Fully extended on the left, random coil on the rigth.

Global comparison between all the posibilities considered for the case of a composition: HPHHPH-PHPHPHHHPHPHHHHHPHPH.

Selection	RC	FE
Stochastic uniform	-10	-9
Remainder	-10	-10
Roulette	-10	-10
Uniform	-8	-9
Tournament	-9	-10

Tabla 1: Comparation of methods in of the solutions obtained in one attempt.

6 Conclusions and further work

In this work I implement a hybrid genetic algorithm for the protein folding problem in 2D HP model, based on works like the Unger and Moult ones [22].

The algorithm doesn't have a great preformance but for short proteins it usually finds properly their naturalized states, how we can see in the results exposed in the previous section.

Large proteins only composed by Hidrophobic aminoacids are also well suited. For long chains the algorithm is too slow and has more problems to achieve the native state.

The method of selection which seems to work better in this case is the reminder selection. It could be due the rendomness in the mutation process and a too high temperature parameter in the selection which allows flexibility.

In future works the intention could be to speed up the program, implement some other method and make more comparisons between them. Also, one could try to compare different values of the posible tunable parameters.

References

[1] Nelson, Phillip *Biological Physics* pag. 50 Ed W.H. Freeman and Company, NY. 2008

[2] Editorial: *So much more to know* Science 2005, 309:78-102

[3] Ken A Dill, S Banu Ozkan, Thomas R Weikl, John D Chodera, Vincent A Voelz, *The protein folding problem: when will it be solved?*, Current Opinion in Structural Biology, Volume 17, Issue 3, June 2007, Pages 342-346,

[4] Levinthal, Cyrus (1969) *How to Fold Graciously*. Mossbauer Spectroscopy in Biological Systems: Proceedings of a meeting held at Allerton House, Monticello, Illinois: 22–24.

[5] Bonnie Berger and Tom Leighton. 1998. *Protein folding in the hydrophobic-hydrophilic (HP) is NP-complete.* (RECOMB '98), Sorin Istrail, Pavel Pevzner, and Michael Waterman (Eds.).

[6] Kit Fun Lau and Ken A. Dill. *Dominant Forces in Protein Folding*. Biochemistry, 29(31):7133–7155, 1990.

[7] Kit Fun Lau and Ken A. Dill. A Lattice Statistical Mechanics Model of the Conforma- tional and Sequence Spaces of Proteins. Macromolecules, 22:3986–3997, 1989.

[8] Dill K.A. *Theory for the folding and stability of globular proteins.* Biochemistry 24, 1985

[9] Thachuk, Chris; Shmygelska, Alena; Hoos, Holger *A replica exchange Monte Carlo algorithm for protein folding in the HP model* BMC Bioinformatics. 2007; 8: 342. Published online 2007 September 17

[10] Alena Shmygelska and Holger H Hoos. *An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem.* BMC Bioinformatics. doi:10.1186/1471-2105-6-30

[11] Ron Unger and John Moult. *Finding the lowest free energy conformation of a protein is an NP-hard problem: Proof and implications.* Bulletin of Mathematical Biology, 55(6):1183–1198, 1993.

[12] *Folding@home* http://en.wikipedia.org/wiki/Folding@home

[13] Firas Khatib, Seth Cooper, Michael D. Tyka, Kefan Xu, Ilya Makedon, Zoran Popović, David Baker, and Foldit Players *Algorithm discovery by protein folding game players* PNAS 2011 ; published ahead of print November 7, 2011, doi:10.1073/pnas.1115898108

[14] G. Mauri, A. Piccolboni, G. Pavesi. *Approximation Algorithms for Protein Folding Prediction.* Recomb 99 Conference, Lyons, France 1999.

[15] Alena Shmygelska, Rosalía Aguirre Hernández, and Holger H. Hoos. 2002. *An Ant Colony Optimization algorithm for the 2D HP Protein Folding Problem* (ANTS '02), Marco Dorigo, Gianni Di Caro, and Michael Sampels (Eds.). Springer-Verlag, London, UK, UK, 40-53.

[16] Alena Shmygelska and Holger H. Hoos. 2003. *An improved ant colony optimisation algorithm for the 2D HP protein folding problem.* (AI'03), Yang Xiang and Brahim Chaib-Draa (Eds.). Springer-Verlag, Berlin, Heidelberg, 400-417.

[17] John H. Holland. *Outline for a Logical Theory of Adaptive Systems.* Journal of the ACM, 9(3):297–314, 1962.

[18] Wolpert, D.H., Macready, W.G. *No Free Lunch Theorems for Search*, Technical Report SFI-TR-95-02-010, (Santa Fe Institute), 1995.

[19] Bäck, T., Fogel, D., Michalewicz, Z. *Evolutionary Computation I*, Institute of Physics publishing, 2000

[20] Thomas E. Creighton. *Experimental Studies of Protein Folding and Unfolding*. Progress in Biophysics and Molecular Biology, 33:231–297, 1978.

[21] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. *Optimization by Simulated Annealing.* Science 220 (4598): 671–680. 1983

[22] Ron Unger and John Moult. *Genetic Algorithms for Protein Folding Simulations.* Journal of Molecular Biology, 231:75–81, 1992.