An Enhanced Genetic Algorithm for Protein Structure Prediction Using the 2D Hydrophobic-Polar Model

Heitor S. Lopes, Marcos P. Scapin

Bioinformatics Lab. (CPGEI), Centro Federal de Educação Tecnológica do Paraná,
Av. 7 de setembro, 3165 – 80230-901 Curitiba, Brazil
hslopes@cpgei.cefetpr.br, mpscapin@cpgei.cefetpr.br

Abstract. This paper presents an enhanced genetic algorithm for the protein structure prediction problem. A new fitness function, that uses the concept of radius of gyration, is proposed. Also, a novel operator called partial optimization, together with different strategies for performance improvement, are described. Tests were done with five different amino acid chains from 20 to 85 residues long and better results were obtained, when compared with those in the current literature. Results are promising and suggest the suitability of the proposed method for protein structure prediction using the 2D HP model. Further experiments shall be done with longer amino acid chains as well as with real-world proteins.

1 Introduction

A protein is a chain of amino acid residues that folds into a specific native 3-dimensional structure under natural conditions, just after being synthesized in the ribosome. The task of predicting this 3-D structure is called the protein structure prediction problem (PSP) and its resolution is of great importance for modern molecular biology.

Exhaustive search of the entire conformational space of a protein is not possible, even for the small ones. Simplified models, where amino acids are laid on a 2- or 3-dimensional lattice, have been proposed. Again, such models are feasible only for small proteins, due to its NP-completeness [1]. Consequently, heuristic optimization methods seem to be the most reasonable algorithmic choice to solve PSP, and, amongst them, many evolutionary computation approaches have been proposed [2], [3], [4], [5], and [6]. In this paper we present an improved genetic algorithm for PSP. Its most important feature is a new fitness function capable of directing the search towards good protein conformations. Using a benchmark, results show that our implementation achieves optimal or quasi-optimal solutions for small proteins.

2 2D HP Model

The 2D HP (2-dimensional Hydrophobic-Polar) model was introduced by [7] and it is the most widely studied discrete model for protein folding in the recent literature. It models the concept that the major contribution to the free energy of the native conformation of a protein is due to interactions among hydrophobic residues. They tend to form a core in the protein structure while surrounded by hydrophilic residues that interface to the environment.

In the HP model, the 20 standard amino acids are divided into two types, according to its affinity to water: hydrophobic (H for non-polar) or hydrophilic (P for polar). As it is a lattice model, the amino acid chain is embedded in a 2- or 3-dimensional square lattice and the movements of the chain are restricted to angles of 90^o. In a legal conformation, the adjacent residues in the sequence must be adjacent in the lattice and each lattice point can be occupied by only one residue.

The free energy of a conformation is inversely proportional to the number of hydrophobic non-local bonds (or H–H bond). An H–H bond occurs if two hydrophobic residues occupy adjacent grid points in the lattice but are not consecutive in the sequence. Each such interaction contributes with –1 to the energy value.

3 Implementation

In this section, we describe in details the application of a genetic algorithm (GA) and the strategies proposed to improve its performance.

3.1 Chromosome Encoding

The dynamics and effectiveness of a GA is strongly influenced by the way solutions are represented. There are two ways of representing a chain in a lattice: either using absolute or relative coordinates. In the former, every amino acid uses Cartesian coordinates to define its position in the lattice. In the latter, the definition of an amino acid position takes into account the position of the previous one, with relative movements. Based on the results presented by [8], our implementation uses internal coordinates. Due to the 2-dimensional lattice used, there are only three possible moves, regarding the previous amino acid of a chain: (R)ight, (L)eft and (F)orward. These moves indicate that the next amino acid of the chain will be folded (together with the remaining forward chain) 90 degrees to the right, to the left or the chain will be stretched ahead.

Therefore, the GA will have a population of individuals with a single chromosome, each one representing a complete conformation. The chromosome is composed by a number of genes corresponding to the number of amino acids in the chain minus one (the starting amino acid of the chain), and every gene is defined over the alphabet $\{R, L, F\}$.

3.2 Initial Population

In this problem, a constraint to be handled is related to the self-avoidance of a conformation, i.e., whether illegal conformations are allowed during evolution or not. If not, it is necessary a procedure that guarantees the generation of only legal conformations in the initial population and in the application of the genetic operators. Another approach, called penalty method, allows the existence of unfeasible conformations during the evolution, but a penalty is added (to the fitness value of the individual) for every lattice point at which there is a collision of more than one amino acid. Our implementation uses the penalty method.

According to [3], the encoding in relative internal coordinates exhibits the problem that initial populations (randomly created) tend to have an increasing number of collisions as the length of the protein increases, making the GA waste efforts with illegal conformations before promising conformations can be found. Based on this statement, a different strategy was used to create the initial population aiming to minimize the collisions while generating a larger initial genetic diversity. This strategy divides the population into two parts that are generated differently. The proportion of each part is established by a user-defined parameter called *PopIniFull*. The first part of the population is randomly generated, as usual, and this is the part that the percentage of the *PopIniFull* parameter indicates. The second part is generated considering each individual as totally unfolded and then applying a number of random mutations between 3 and the total number of genes in the chromosome, uniformly distributed. Using this method, there will be a certain amount of individuals having few mutations that increases the diversity of the initial population and allows that the unfolded parts of the individuals help the evolution process.

3.3 Fitness Function

To evaluate an individual, it is necessary to translate its genotypical encoding, defined over the alphabet {*R, L, F*}, to obtain its Cartesian coordinates. This procedure allows knowing how the amino acids are disposed in the lattice, and then, the computation of an objective goodness measure of the conformation. In this work, we propose a new fitness function composed of three terms, as shown in Equation 1:

$$Fitness = NLB_H \times RG_H \times RG_P \quad \textbf{(1)}$$

where NLB_H is the number of hydrophobic non-local bonds of the conformation and RG_H and RG_P are terms computed using the radius of gyration of the hydrophobic and hydrophilic residues, respectively, as explained below. The product of all terms in this equation indicates that all of them should be maximized.

3.3.1 Hydrophobic non-local bonds

It is believed that hydrophobic non-local bonds are the main force that drives the protein folding process. We are considering the problem as the maximization of the number of H–H bonds thus, for every hydrophobic non-local bond, NLB_H is added by 1. Since we are using a penalty method, NLB_H is decreased whenever a collision oc-

curs. The penalty term, decremented from NLB_H, is composed by the number of grid points which are occupied by more than one residue, multiplied by the penalty weight which, in turn, is set according to the chain length: the longer the chain, the higher it is.

3.3.2 Radius of gyration

The original HP model uses only the hydrophobic non-local bonds term to evaluate an individual but, according to [8], without a modified energy function, there will exist large plateaus in the energy landscape on which local search cannot find a descent direction, leading to a random search. This fact was also experienced in our preliminary implementation and, aiming to avoid this trap and enhance the fitness function, we propose the use of a new concept, called radius of gyration (*RG*).

RG of a solid body is the radial distance from a given axis at which the mass of a body could be concentrated without altering the rotational inertia of the body relative to that axis [9]. Hopefully, using *RG* in the fitness function the fitness landscape can be changed in such a way that the fitness function rewards more compact conformations with the same number of H–H bonds, bringing the evaluation closer to reality.

RG, in the scope of the PSP, indicates how compact a set of amino acids is: the more compact a conformation, the smaller is its radius of gyration. In this term of the fitness function, only hydrophobic residues were considered, rewarding the conformations that have smaller values of radius of gyration. This term is presented in Equation 2:

$$RG_H = MaxRG_H - \sqrt{\frac{\sum_{i=1}^{NH}\left[(x_i - \overline{X})^2 + (y_i - \overline{Y})^2\right]}{NH}}, \tag{2}$$

where x_i and y_i are the Cartesian coordinates of the i-th hydrophobic residue, $\overline{X}$ and $\overline{Y}$ are the mean values of all hydrophobic x_i and y_i, respectively; NH is the number of hydrophobic residues in the chain; and $MaxRG_H$ is the radius of gyration of the amino acid chain totally unfolded. The second part of Equation 2 represents the radius of gyration of hydrophobic residues related to the point given by the mean coordinates, and it is subtracted from $MaxRG_H$ in order to maximize RG_H.

The term related to the hydrophilic radius of gyration in the fitness function has the opposite purpose as RG_H: it fosters the spreading of hydrophilic residues towards the edge of the conformation. This term is calculated in the same way as in Equation 2, except that, in this case, only hydrophilic residues are considered, and it is not subtracted from any other value (as in Equation 2). Using RG_H computed before, RG_P can be obtained using Equation 3:

$$DIFRG = \sqrt{\frac{\sum_{i=1}^{NP}\left[(x_i - \overline{X})^2 + (y_i - \overline{Y})^2\right]}{NP}} - RG_H \quad , \qquad (3)$$

$$RG_P = \begin{cases} 1 & if\ DIFRG \geq 0 \\ \dfrac{1}{1 - DIFRG} & otherwise \end{cases}$$

In Equation 3, *DIFRG* computes the difference between the hydrophilic and the hydrophobic radii of gyration. A positive difference for *DIFRG* means that the hydrophobic residues are buried inside the conformation, while the hydrophilic ones are outside. Such situation is desired and in this case, the hydrophilic radius of gyration has no influence in the fitness function. However, if the opposite is true, meaning that the hydrophobic residues are more spread than the hydrophilic, which is not desired, this conformation will be penalized, decreasing its fitness value.

3.4 Genetic Operators and Local Improvement Strategies

In GA, genetic operators are used to create new individuals by means of modifying existing ones. Therefore, it is necessary a method for choosing individuals from the current population in order to apply the genetic operators. We used the tournament selection method that randomly selects a number of individuals from the population. These individuals compete in a tournament and the best one is chosen for the application of the operators. The first operator that is applied during the generation of a new population is the crossover operator. For this problem, this operator plays an important role since a piece of structure (conformation) that has been adequately folded can be of further use in the construction of a complete solution [10]. Two types of crossover were implemented: 1- and 2-point crossover and both are applied with the same probability during the evolution.

Another operator commonly used in GA is mutation. In this work, two different types of mutation were developed. The first is the simple mutation where each gene is tested, according to the mutation probability, to verify whether the actual value of the gene will be changed or not. The second type, called Improved Mutation, works as the simple mutation except by the fact that after each mutation is applied, the individual is reevaluated to check if its fitness has increased. In this case, the change is maintained, otherwise it is discarded. In order to guarantee some diversity during the evolution, 40% of the mutations are simple and 60% are improved mutations.

In our implementation, both crossover and mutation probabilities are not fixed during generations. They have an initial and a final value, respectively for the first and the last generation. The exact probability value in a given generation is a linear interpolation of the initial and final values.

A specially devised operator used in this work is named Partial Optimization. The basic idea of this operator is to randomly select two non-consecutive residues of the protein and fix their position in the lattice. Then, all the different possibilities of locating the intermediate residues maintaining the connectivity of the chain are calcu-

cating the intermediate residues maintaining the connectivity of the chain are calculated. The conformation that gives the maximum fitness among all of them is kept. This operator was inspired in a generalization of the 2-opt heuristics proposed by [11] for the traveling salesman problem. The number of intermediate residues to be permutated is a user-defined parameter named Partial Optimization size.

In preliminary tests, the GA frequently got trapped in local minima. Thus, it was necessary to implement a strategy, called Decimation, to make the GA overrides this situation. After each generation, the fitness of the best individual is checked in order to verify whether or not it has changed from the previous generation. If not, a counter is increased by 1. If so, the new best fitness is kept and the counter is reset to 0. When the non-improvement counter reaches 10, the decimation strategy is applied. The idea is to eliminate all individuals of the current population, except the best, and generate again a new population (in the same way explained in section 3.2), including the best individual previously found. Applying this strategy makes the population to have a large genetic diversity, hopefully allowing further evolution. A point that needs to be taken into account is the fact that all the newly generated population probably will have very low fitness values compared to the best individual previously found. Therefore, it is necessary to decrease the selective pressure giving more chance to all individuals to be selected. This is done by decreasing the tourney size in the selection method at the same time that the probability of applying the Improved Mutation is increased. This strategy decreases competitiveness between individuals and permits that all the population becomes, on average, a little better and contributes to the evolution. When this strategy is applied, the non-improvement counter returns to 0 and the verification of the best fitness change proceeds until the last generation.

4 Computational Experiments and Results

Several experiments were performed with the same instances used in [12], for five amino acid chains with 20, 36, 48, 64, and 85 residues. Such instances are not real-world proteins, but a benchmark for which the optimal folding with the 2D HP model is known. Despite of this, it would be interesting to evaluate our method comparing it with a similar one, over the same instances. According to [12], the maximum number of H–H bonds for those instances are: 9, 14, 23, 42, and 52, respectively.

For all the experiments, the parameter set used is shown in Table 1. It was not done a combinatorial experiment so as to find the most efficient set of parameters within the possible range. Instead, we conducted some preliminary tests with different combinations of parameters using a single instance. The set of parameters that performed best among the combinations tested was chosen as default. It is worth to note that possibly another set of parameters could perform better than those used here, but this investigation is subject of future research.

As mentioned before, the penalty weight was (empirically) set according the length of the chain: 2, 2.5, 3, 3.5 and 4 for the 20-, 36-, 48-, 64- and 85-residue chains, respectively.

Tests were run 100 times and the individual with the highest number of hydrophobic non-local bonds from the last generation was considered the best of the run. The

overall best individual for each instance is shown in Table 2, together with the number of times this solution was found within 100 runs. The mean number of H–H bonds of the 100 best individuals was calculated and also presented in that table, together with the results obtained by [12], for the purpose of comparison. Values in bold represents the best solutions.

Table 1. Set of parameters for the genetic algorithm

Parameters	**Values**
Population size	500
Number of Generations	100
PopIniFull	30%
Tourney size	3%
Elitism	Yes
Crossover probability (initial / final)	50% / 70%
Mutation probability (initial / final)	5% / 10%
Partial optimization probability	4%
Partial optimization size	7 residues

Table 2. Comparison of results. Numbers in parenthesis indicates how many times the best score was found in 100 different runs and the bold values indicate the best result for a given instance

Chain length	König and Dandekar [12]		Our implementation	
	Best solution	Mean value	Best solution	Mean value
20	9 (100×)	9.00	9 (100×)	9.00
36	14 (8×)	12.40	14 (6×)	**12.44**
48	23 (1×)	18.50	23 (2×)	**20.06**
64	37 (1×)	29.30	**40** (1×)	**33.58**
85	46 (1×)	40.80	**51** (2×)	**45.74**

For the 20-residue chain, as the global minimum was always reached, the performance measure considered was the mean number of energy evaluations needed to find the global minimum. König and Dandekar's implementation needed an average of 11824 energy evaluations while ours took 10830.

For the first three chains (namely, 20, 36 and 48 amino acids chains) our results were very similar to [12]. Both implementations were able to find the global optimum but, in average our implementation performed better for the 36- and 48-long amino acid chains. For the 64- and 85-long amino acids chains, our implementation obtained much better results than [12], either considering the best result or the mean value of energy function. For both instances, our best result was very close to the optimal solution known (42 and 52 H–H bonds, respectively).

In general, our GA got similar results to [12] for the smaller chains and better results for the longer chains. It is important to consider that a difference of one bond from a conformation to another indicates a great improvement obtained by the algorithm and jumping from the closest local minimum to the global minimum can be

considered a great achievement. From a solution to another with single bond more, it can mean a quite different folding.

The two best results found for the 85-residue chain are presented in Fig. 1, where the black dots are the hydrophobic residues and the white dots, the hydrophilic. The biggest dot is the beginning of the chain.

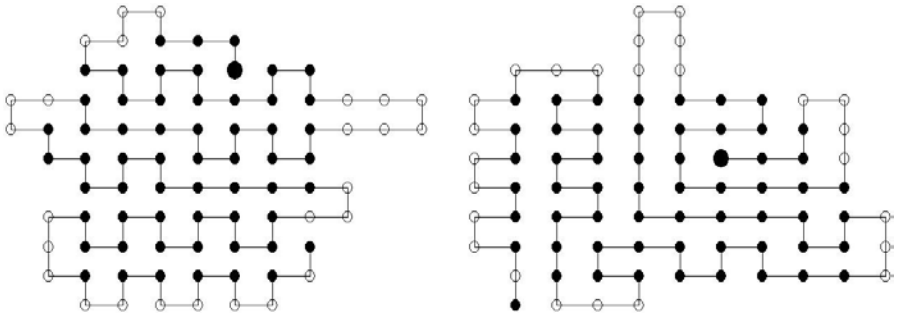

Fig. 1. Best conformations found for the 85-long amino acid chain

5 Conclusions

This paper presented novel strategies for using a genetic algorithm for the protein structure prediction problem using the 2D HP model. The use of the concept of radius of gyration in the fitness function took some smoothness to the fitness landscape, allowing better solutions to be found. Using this fitness function, two conformations with the same number of H–H bonds can be adequately discriminated. Also, the use of the partial optimization and improved mutation operators, together with the decimation strategy have enhanced the GA, allowing it to escape from local minima.

Besides the enhancements in the GA, it is important to emphasize the results obtained. While for short chains the results got no significant improvements (compared with [12]), for the long ones, significant local minima were found, suggesting that there is room for further improvement with longer chains. This subject shall be addressed in further experiments.

The two different solutions shown in Fig. 1 emphasize the difficulty of the PSP problem using a lattice model. The use of this model and the energy function based on the number of H–H bonds implicitly implicates a (strongly) multimodal fitness landscape with many equal-sized plateaus. This fact, by itself, requires efficient search strategies specially when using evolutionary computation techniques.

Exhaustive experiments aiming to find the best parameter set for the GA were not performed, even though the results achieved were very promising. Finding such set of parameters is computationally intensive and care must be taken on its generalization. Experience suggests that not only the size of the amino acid chain is important, but also, some implicit characteristic of the folded structure. These research directions will be explored in the near future.

Overall, results encourage the continuity of the work towards a more complex lattice model, and further tests with the use of real-world protein sequences.

6 Acknowledgments

Authors would like to thank CNPq for the financial support for H.S.Lopes (grants 305720/04-0 and 402018/03-6) and M.P.Scapin (grant 131355/04-0).

References

1. Berger, B., Leight, T.: Protein Folding in the Hydrophobic-hydrophilic (HP) Model is NP-Complete. J. Comp. Bio. 5 (1998) 27–40
2. Unger, R., Moult, J.: A Genetic Algorithm for Three Dimensional Protein Folding Simulations. In: Proceedings of the 5th Annual International Conference on Genetic Algorithms (1993) 581–588
3. Patton, A.L., Punch III, W.F., Goodman, E.D.: A Standard GA Approach to Native Protein Conformation Prediction. In: Proceedings of the 6th International Conference on Genetic Algorithms, Morgan Kauffman (1995) 574–581
4. Pedersen, J.T., Moult, J.: Protein Folding Simulations With Genetic Algorithms and a Detailed Molecular Description. J. Mol. Biol. 269 (1997) 240–259
5. Krasnogor, N., Pelta, D., Lopez, P.E.M., Canal, E.: Genetic Algorithm for the Protein Folding Problem: a Critical View. In: Proceedings of Engineering of Intelligent Systems (1998) 353–360
6. Day, R.O., Lamont, G.B., Pachter, R.: Protein Structure Prediction by Applying an Evolutionary Algorithm. In: International Parallel and Distributed Processing Symposium (2003) 155–162
7. Dill, K.A.: Theory for the Folding and Stability of Globular Proteins. Biochemistry 24 (1985) 1501–1509
8. Krasnogor, N., Hart, W.E., Smith, J., Pelta, D.A.: Protein Structure Prediction with Evolutionary Algorithms. In: Proceedings of the International Genetic and Evolutionary Computation Conference (1999) 1596–1601
9. Beer, F.P., Johnston, E.R.: Vector Mechanics for Engineers – Statics. New York: McGraw Hill, 1980
10. Unger, R., Moult, J.: On the Applicability of Genetic Algorithms to Protein Folding. In: 26th Hawaii International Conference on System Sciences, vol. I, IEEE Press (1993) 715–725
11. Croes, G.A.: A Method for Solving Traveling Salesman Problems. Oper. Res. 5 (1958) 791–812
12. König, R., Dandekar, T.: Improving Genetic Algorithms for Protein Folding Simulations by Systematic Crossover. BioSystems 50 (1999) 17–25