A Multi-objective Genetic Algorithm for the Protein Structure Prediction

Alfonso E. Márquez Chamorro*, Federico Divina, Jesús S. Aguilar-Ruiz and Gualberto Asencio Cortés
School of Engineering,
Pablo de Olavide University, Seville, Spain
* *Email: amarcha@upo.es*

***Abstract*—The Protein Structure Prediction (PSP) problem consists of predicting the structure of a protein from its amino acids sequence, and have received much attention lately. In fact, being able to predict the structure of a protein, would allow to know the function of the protein. In this paper, we propose a multi-objective evolutionary algorithm for the PSP problem. The prediction model consists of a set of rules that determine possible contacts between amino acids. Such rules are based on four specific amino acid properties, which are involved in the folding process: hydrophobicity, polarity, net charge and residue size. In order to increase the interpretability of the results, rules are organized in a $20x20$ matrix where each cell contains the specific rules for a possible pair of residues. The high accuracy values obtained confirm the validity of our proposal.**

***Keywords*-Protein Structure Prediction; Contact Map; Evolutionary Computation; Multi-objective Optimization; Amino Acid Properties**

I. Introduction

The problem of predicting the tertiary structure of a protein from its amino acids sequence is called Protein Structure Prediction (PSP), and it is one of the main open problems in Computational Biology. In fact, the 3D structure of a protein determines its function and being able to predict how a protein will fold, would have a huge impact in numerous fields, e.g., new drugs design. There are some experimental methods, like NMR and X-ray crystallography, that can determine the 3D structure of a protein. However such methods are very expensive, both financially and in terms of time. Therefore, computational methods are needed as they could provide a faster and economic way to tackle the PSP problem.

In particular, soft computing methods seem particularly suited for solving this problem, as they can deal with noise and uncertainty in the data. Different soft computing paradigms, e.g., neural network [1] or support vector machines [2], were applied to the PSP problem. PSP can be considered as a search problem, where the search space is determined by all possible rules involved in the folding process of a protein. Such a space is huge, and very complex. A soft computing paradigm that can obtain good results in these cases, is represented by Evolutionary Algorithms (EAs). This is because EAs are characterized by very good search capabilities and have the ability to escape from local optima. An example of application of an EA to PSP is [3], where a torsion angles model was developed. HP Model and Lattice Model were performed as first PSP evolutionary approaches [4] and nowadays are still used [5]. A contact map generator was included in [6]. [7] used a binary encoding of the protein data. All these methods consider PSP as single-objective optimization problem. This means that all the objectives that have to be optimized are combined into a single fitness function, that guide the evolutionary search performed by the EA. In particular, these methods use a single-objective potential energy function. The function is used in order to find the three-dimensional native conformation with minimum energy from a protein sequence of amino acids.

The single objective approach works well when there is only one objective to optimize, or all the objectives are not in conflict with each other. Often, however, a problem requires to optimize different objectives at the same time, and it can be difficult to combine them in a single function. Moreover, in this case, the search space is highly complex and in such a scenario, it is often impossible to find a single optimal solution. Instead, one is usually more interested in finding a set of solutions that presents a good compromise among all the objectives. Such solutions are generally denoted as the Pareto set, because based on the notion of Pareto dominance. Rather than combining the multiple objectives into a single fitness function, a better approach to find this optimal set is to optimize the objectives separately, i.e., treat the problem as a multi-objective problem. EAs are particularly suited for tackling multi-objective optimization problems, mainly due to the population-based nature of EAs. This allows the generation of several elements of the Pareto set in a single run. Some of the best known multi-objective EAs (MOEAs) are NSGA, SPEA, NSGA-II, SPEA-II and PAES-II [8].

Several prediction methods have considered PSP problem as a multi-objective optimization problem (MOP). [9] developed MI-PAES as a modified version of PAES using a torsion angles model. A parallel multi-objective optimization was performed by using Chemistry at HARvard Macromolecular Mechanics (CHARMM) energy function in [10]. [11] proposed a multi-objective Feature Analysis and Selection Algorithm (MOFASA) in order to solve the Protein Fold Recognition (PFR) problem. In [12], a I-PAES algorithm is used as search procedure for exploring the space of the PSP problem. The concept of bond and non-bond

energies are included in the fitness function of this approach.

In this work, a MOEA is proposed for the protein contact map prediction problem. In particular, we present a multi-objective scheme based on the Strength Pareto Evolutionary Algorithm (SPEA) for the PSP problem. The main novelty of our proposal, is that the prediction is based on a set of amino acid properties. The reason for basing the prediction on such properties, is that it has been shown that amino acids that are in contact, are characterized by same properties. None of the cited algorithms consider amino acid properties for the prediction.

The prediction model consists of a set of rules which determine possible contacts between amino acids with some specific requirements. We obtain a $20x20$ matrix where each cell represents a possible pair of residues and the cited rules are classified inside each corresponding cell. In our proposal, from this matrix, we build a contact map representation which is an approximation of the three-dimensional structure of the protein.

We can define a contact map as a matrix C of size NxN, where N is the length of the residue sequence and where each cell C_{ij} represents an amino acid pair (i, j). We assign to each cell 1 or 0 depending on there is or not a contact between amino acids i and j. A contact is established if the distance between them is lower than a determined threshold μ, expressed in angstroms.

This work is organized as follows. Our evolutionary approach is described in section II. Section III presents the experimentation and obtained results. Finally, section IV, includes some conclusions and possible future works.

II. Methodology

In this section we will explain the main characteristics of our algorithm. The experimental procedure is represented in Figure 1. As it can be seen, we obtain a set of proteins from Protein Data Bank (PDB) [13] as input data of our MOEA. The output of the algorithm is represented by a set of rules to determine residue-residue contacts. These rules are organized into a $20x20$ matrix M. Such a matrix contains an entry for each possible pair of amino acids (X, Y), and each entry contains the rules relative to (X, Y). Finally, we obtain the protein contact map for a new protein by applying the cited rules of M.

Before describing the actual algorithm, in the next section, we provide the basic notions of multi-objective optimization.

A. Multi-objective Optimization

As already mentioned, there exist problems where there are different objectives that need to be optimized at the same time, and that are often in conflict with each other. In such cases, we talk about multi-objective optimization (MOP).

In such problems, we want to optimize a set of objective functions $(f_1(x), f_2(x)...f_n(x))$, where $x = (x_1, ..., x_p)$ is a vector of real parameters or decision variables belonging

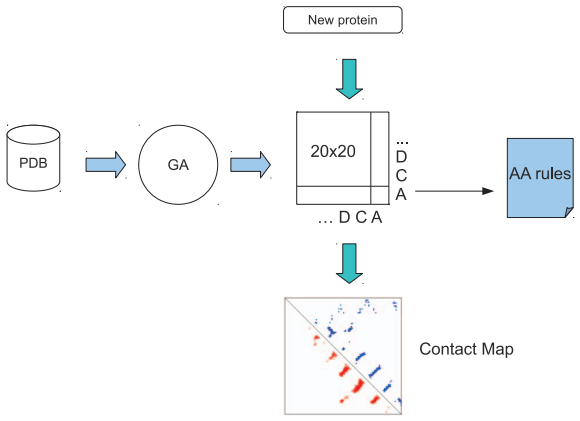

Figure 1: Experimental procedure scheme.

to a universe X and $f_i(x)$ are arbitrary linear or nonlinear functions. Therefore, the problem consists of finding the x that provides the best compromise value for all $f_i(x)$.

In MOP problems it is usually not possible to find a single optimal solution. Instead, one is usually interested in finding a set of solutions that present a good trade-off among the different objectives. The solution then consists of a set of elements. Such a set is called the Pareto front. In order to identify this set of solutions, we need the notion of dominance. A solution x is said to be not dominated iff there is not another solution y such that: $f_i(y) <= f_i(x)$ for all $i = 1, .., n$ and $f_i(y) < f_i(x)$ for some i. The Pareto front is formed by all the non-dominated solutions.

The PSP problem can bee seen as a MOP, since the problem involves more than one objective function. In the other cited approaches, the goal is to find a set of optimal conformations with minimum free energy that optimize all the possible objectives of the folding process.

In the following, we provide the details of the MOEA used in this paper.

B. The proposed MOEA

In our approach, the objectives to be optimized are the recall and the precision.

Each individual of the population encodes a rule that predicts whether or not two amino acids i and j are in contact.

Rules are based on four different physico-chemical properties of the amino acids, being these properties the hydrophobicity, the polarity, the net charge and the residue size. We have selected these particular four properties since it has been proven that they are involved in the folding process of a protein, see [14]. In a previous study [15], we have carried out several experiments and statistical tests aimed at testing the validity of this selection of properties.

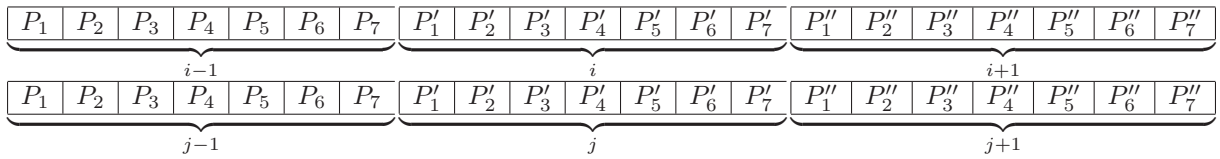

Figure 2: Example of an individual for the amino acids $i-1$,i,$i+1$,$j-1$,j,$j+1$.

Each rule represents two windows, where the amino acids that precede and follow i and j are considered. The first windows is relative to amino acids $i-1, i, i+1$, while amino acids $j-1, j, j+1$ are represented in the second window, as can be seen in figure 2, where i and j are two possible amino acids in contact. Each amino acid is represented by seven real value genes. Positions P_1, P_2 represent the range of hydrophobicity according to the Kyte-Doolittle scale [16]. Positions P_3, P_4 represent an interval of values of polarity in the Grantham profile [17]. P_5 indicates the net charge value (-1,0 and 1 for negative, neutral and positive charge respectively) of the residue according to Klein scale [18]. Finally, P_6 and P_7 represent the range related to the size or volume of the residues using Dawson scale [19]. Hydrophobicity and polarity values were normalized between -1 and 1 and the residue size values were normalized between 0 and 1.

The aim of the algorithm is to find precise and general rules for identifying residue-residue contacts. In a previous work [15], we have proposed a single-objective EA for the PSP problem, where the F-measure was used as fitness function. In this paper, we aim at finding the best compromise between the correctly identified contacts rate (recall) and the real predicted contacts rate (precision) of our algorithm. The formula of F-measure is shown in the Equation 1.

$$F_{measure} = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision}. \quad (1)$$

The algorithm proposed in this paper, is based on SPEA [20]. This algorithm is based on the selection of all non-dominates solutions or Pareto Front (in this case, according to their recall and precision values) at every generation and the fitness of these individuals is based on the number of solutions they dominate.

The algorithm starts by randomly initialize the population. After individual are evaluated by first computing the values of the objectives, and then by evaluating the dominance ranks. Elements are selected using a binary tournament selection mechanism.

Offspring are generated with both crossover and mutation. A two-points crossover operator is used with a give probability. The mutation operator used first randomly selects a gene of the individual. If the gene encodes the charge of an amino acids, then its values is randomly changed to another of the allowed values. Otherwise, the value is decreased or increased by 0.1. After that mutation has been applied, the individual is checked for validity. If the new individual is an invalid rule, the mutation is discarded.

The algorithm is run for a maximum number of generations, and after that, the set of non-dominated rules are extracted and returned as result. However, if the fitness of the best individual does not increase over twenty generations, the algorithm is stopped. The algorithm can be run various times, in order to find more rules and in this way to improve the results. In this case, after each iteration of the algorithm, the extracted rules that increase global F-measure of the final solution, will be included to the final solution. Repeated or redundant rules are discarded. Thus, the more runs of the EA, the more rules will be added to the final solution.

Table I: Parameter setting used in the experiments.

Populazion size	100
Crossover probability	1.0
Mutation probability	0.5
Max. number of generations	100
Tournament size	2

III. Experiments and results

In order to test the effectiveness of the proposal we have selected from PDB a protein data set taken from [1].

This data set consists of a set of 65 proteins, whose sequence identity is lower than 25%. The maximum length of the protein sequence is 100 residues. From these proteins, we have extracted their amino acid sequences and have calculated the distances $d(i,j)$ between each pair of residues (i,j) using the Euclidean distance equation $d(i,j) = \sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}$, where (x_1,y_1,z_1) represent the atomic coordinates of the first amino acid and (x_2,y_2,z_2) are the coordinates of the second amino acid. C_β-C_β distances (C_α for Glycine) were taken into account for the calculations. A threshold of 8 angstrom (Å) is established to determine a contact. In order to avoid the effect of learning local contacts, we set a minimum sequence separation of 7 residues between each pair of amino acids to establish a contact. All these requirements were found in [1]. A 10-fold cross-validation were performed during the experimentation.

The parameter setting of the algorithm are shown in table I. These values were determined after having run several preliminary runs of the algorithm.

The optimal number of rules for the prediction is unknown. For this reason, we test with a different number of iterations of the algorithm, more specifically 1,2,5 and 10 thousands executions. This means that the final set of rules is incrementally built. Each time the algorithm is run, the set of rules it returns are added to the final solution set, as described in section II.

Two measures, frequently used in literature for the PSP problem, were calculated to test the validity of our proposal: Coverage and Accuracy.

- Coverage, represents the percentage of correctly identified contacts C/C_t, where C is the number of correctly predicted contacts of a protein and C_t is the total number of contacts of the protein. This measure is also called Sensitivity or Recall.
- Accuracy, represents the percentage of real predicted contacts C/C_p, where C_p is the number of predicted contacts.

In table II we can observe the obtained results. The first column indicates the number of executions and the second and third ones indicate the average Coverage and Accuracy respectively. The fourth column represents the number of resulting rules. Standard deviation is also reported in the table.

Table II: Average values and standard deviation of the obtained results.

Runs	$Coverage_{\mu\pm\sigma}$	$Accuracy_{\mu\pm\sigma}$	# rules
1,000	$0.11_{\pm 0.03}$	$0.28_{\pm 0.15}$	2,221
2,000	$0.15_{\pm 0.10}$	$0.27_{\pm 0.12}$	4,548
5,000	$0.17_{\pm 0.15}$	$0.24_{\pm 0.13}$	11,560
10,000	$0.19_{\pm 0.16}$	$0.23_{\pm 0.13}$	27,137

A maximum value of coverage of 0.19 was obtained for $10,000$ executions and the best value of accuracy (0.28) is obtained for $1,000$ executions. We can observe that the value of the coverage is higher when the number of executions increases. This is due to the higher number of covered cases with a higher number of rules. On the other hand, the accuracy rate decreases in this case. This decrement of the accuracy is due to the higher possibility of error with a higher number of rules.

We can conclude that it is difficult to estimate the correct number of rules for the prediction. In fact, if the number of rules is increased, the recall is also increased but the precision is decremented. In turn, the number of rules varies considerably by increasing the number of executions of the algorithm. On average, we have obtained 2.52 rules per execution.

We have obtained similar values than other approaches in literature. For example, in [1] a maximum accuracy of 0.26 was obtained. In [2], the average value of accuracy was also around 24%.

As already mentioned, all the resulting rules were organized in a $20x20$ matrix. Each cell of this matrix represents a pair of amino acids, and it contains the rules related to these two amino acids. It follows that the same rule can be present in different cells, since a rule indicates several characteristics of amino acids that can be fulfilled by different pairs of residues. When we need to predict the folding of a new protein, we apply the corresponding rules for each pair of residues of the protein sequence and we determine its contact map. An example of a resulting rule is showed in Figure 3. Each position represents a value for a different property as explained in section II and encodes a feature of a possible amino acid. For instance, the hydrophobicity value for the amino acid in position i is between 0.52 and 0.92, the polarity value between -1.0 and -0.93 and neutral charge (0.0). Therefore, this amino acid could be L (Lysine) or F (Phenylalanine), which fulfill all these features according to the cited scales. The features of the position j are fulfilled by amino acid T (Threonine). Thus, this rule will be included in the cell LT and FT of the final solution matrix. For $1,000$ executions, a total of 23 rules were classified in the LT cell and 15 rules in the FT cell. These kind of rules are easily interpreted by an expert in the field.

IV. Conclusions and future work

In this work, we propose a multi-objective evolutionary approach for the PSP problem by predicting protein contact maps. Our algorithm generates a $20x20$ matrix which represents a set of rules for each possible pair of amino acids. These rules are based on physico-chemical properties of amino acids. Whenever the folding of a new protein has to be predicted, we can generate its contact map according to the specific rules of each pair of residues of the protein sequence. These organization facilitates the interpretation and inspection of the rules. We believe this is a very important aspect of the prediction, since it would allow experts in the field to easily inspect the obtained rules for obtaining further knowledge about the protein folding process.

The algorithm was tested on a set of protein sequences that had been previously used in the literature. The algorithm obtained satisfactory results on this dataset, which are similar to the results obtained by other approaches.

As future works, we are planning to include in our approach other significant physico-chemical properties involved in the folding process in order to increase the quality of the prediction model. The variable size of the window of an individual and the separation between two amino acids in contact in the sequence, must be taken into account in next versions of our algorithm. Our algorithm must be validated with a higher number of proteins and with a higher sequence length. It would be interesting to analyze the huge number of resulting rules, by extracting knowledge

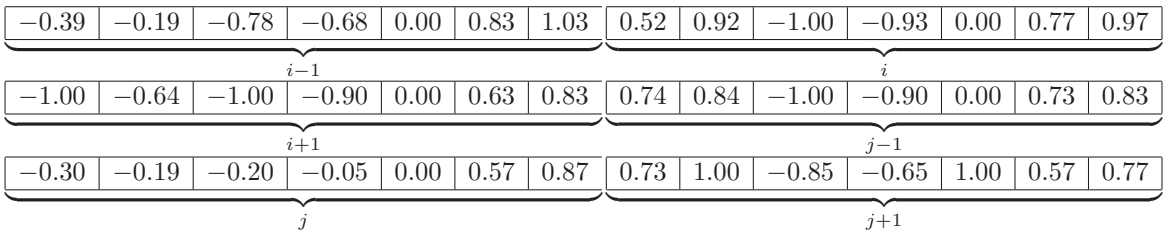

Figure 3: Example of a resulting rule.

as statistics and general conclusions from the final set of rules.

REFERENCES

[1] P. Fariselli, O. Olmea, A. Valencia, and R. Casadio. Prediction of contact map with neural networks and correlated mutations. *Protein Engineering*, 14:133–154, 2001.

[2] J. Cheng and P. Baldi. Improved residue contact prediction using support vector machines and a large feature set. *Bioinformatics*, 8:113, 2007.

[3] Y. Cui, R.S. Chen, and W. Hung. Protein folding simulation with genetic algorithm and supersecondary structure constraints. *Proteins: Structure, Function and Genetics*, 31:247–257, 1998.

[4] R. Unger and J. Moult. Genetic algorithms for protein folding simulations. *Biochim. Biophys.*, 231:75–81, 1993.

[5] T. Hoque, M. Chetty, and A. Sattar. Extended hp model for protein structure prediction. *J Comput Biol.*, 16(1):85–103, 2009.

[6] N. Gupta, N. Mangal, and S. Biswas. Evolution and similarity evaluation of protein structures in contact map space. *Proteins: Structure, Function, and Bioinformatics*, 59:196–204, 2005.

[7] G. Zhang and K. Han. Hepatitis c virus contact map prediction based on binary strategy. *Comp. Biol. and Chem.*, 31:233–238, 2007.

[8] C.A. Coello, D.A. Van Veldhuizen, and Lamont G.B. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002.

[9] MV. Judya, KS. Ravichandrana, and Murugesan K. A multi-objective evolutionary algorithm for protein structure prediction with immune operators. *Comp. Methods in Biomechanics and Biomedical Engineering*, 12(4):407–413, 2009.

[10] JC. Calvo and Ortega J. Parallel protein structure prediction by multiobjective optimization. *Parallel, Distributed and Network-based Processing*, 12(4):407–413, 2009.

[11] SYM. Shi and N. Suganthan. Parallel protein structure prediction by multiobjective optimization. *KanGAL Report*, 2004007:1–7, 2004.

[12] V. Cutello, G. Narzisi, and G. Nicosia. A multi-objective evolutionary approach to the protein structure prediction problem. *J. R. Soc. Interface*, 3:139–151, 2006.

[13] Protein data bank web. *http://www.pdb.org*.

[14] J. Gu and P.E. Bourne. *Structural Bioinformatics (Methods of Biochemical Analysis)*. Wiley-Blackwell, 2003.

[15] AE. Marquez, F. Divina, JS. Aguilar-Ruiz, and G. Asencio. An evolutionary approach for protein contact map prediction. *Lecture Notes in Computer Science*, 6623:101–110, 2011.

[16] J. Kyte and R.F. Doolittle. A simple method for displaying the hydropathic character of a protein. *J. J. Mol. Bio.*, 157:105–132, 1982.

[17] R. Grantham. Amino acid difference formula to help explain protein evolution. *J. J. Mol. Bio.*, 185:862–864, 1974.

[18] P. Klein, M. Kanehisa, and C. DeLisi. Prediction of protein function from sequence properties: Discriminant analysis of a data base. *Biochim. Biophys.*, 787:221–226, 1984.

[19] DM. Dawson. *The Biochemical Genetics of Man*. Brock, DJH., Mayo, O. eds., 1972.

[20] E. Zitzler and L. Thiele. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. *Technical report Computer engineering and Networks Laboratory (TIK)*, 43:1–43, 1999.