"Mircea cel Batran" Naval Academy Scientific Bulletin, Volume XVIII – 2015 – Issue 1
Published by "Mircea cel Batran" Naval Academy Press, Constanta, Romania // The journal is indexed in:
PROQUEST SciTech Journals, PROQUEST Engineering Journals, PROQUEST Illustrata: Technology, PROQUEST Technology Journals, PROQUEST Military Collection PROQUEST Advanced Technologies & Aerospace

PARAMETER OPTIMIZATION OF PSO FOR PROTEIN STRUCTURE PREDICTION PROBLEM IN THE 2D HP MODEL

Andrei BAUTU[1]
Elena BAUTU[2]
[1] Lecturer Ph.D., Department of Navigation and Naval Transport, Naval Academy "Mircea cel Bătrân", Romania, andrei.bautu@anmb.ro
[2] Lecturer Ph.D., Department of Mathematics and Informatics, "Ovidius" University, Romania

Abstract: *Proteins are considered the most important molecules found in living cells because they are fundamental to many of the life processes. In order to accomplish their tasks, proteins fold in their native state, which is the three-dimensional arrangement of their atoms in which the protein reaches its minimum energy. The protein structure prediction (PSP) problem consists in finding the native state of a protein starting from its atoms. The HP model (Hydrophobic-Polar) is one of the simplified folding models that have been used for this problem. Despite its simplicity it captures well enough the interactions of atoms within the molecule. However, the protein folding problem in the HP model is NP-hard both In 2D, and 3D. In previous papers we have applied Particle Swarm Optimization (PSO) to the PSP problem with good results compared to other meta-heuristic methods. In this paper we seek to optimize the parameters of PSO to improve its results for this problem.*

Keywords: *protein structure prediction, particle swarm optimization, genetic algorithms, HP model*

Introduction

Proteins are considered the most important molecules found in living cells because they are fundamental to many of the life processes. In order to accomplish their tasks, proteins fold to their native state which represents the energetic ground state of the protein. This particular three-dimensional arrangement of the atoms of the protein is the state in which the protein reaches its minimum energy. The protein folding problem was included by *Science* magazine in the list of biggest unsolved science problems [1]. Special algoritms had to be invented for protein folding simulations due to the high computational complexity, even with very simplified models.

The protein structure prediction (PSP) problem consists in finding good computational algorithms for prediction of protein native state starting from the amino acid sequence. It is part of the larger protein folding problem and a long standing goal of computational biology. Research on the PSP problem focuses on two main directions: prediction based on existing databases of protein foldings and similarities between proteins, and prediction using physics laws without derived knowledge [2]. Our current research lays on the latter path.

The following section presents briefly the protein folding terminology and the goals of the PSP problem. Section 3 presents the 2D lattice HP model and two folding encodings. Section 4 presents the PSO for the PSP problem. Section 5 presents the exp erimental results obtained with PSO and GAs for various proteins. The conclusions section contains some remarks ab out current status and future work directions.

The Protein Structure Prediction Problem

Proteins perform a vast range of indispensable roles: enzymes speed up chemical reactions, hormones regulate the quantities of molecules, immunological proteins protect the cell of foreign agents, etc. From the chemical p oint of view, they are organic compounds created from sequences of amino acids. Each amino acid has a structure common to all amino acids and a residual attachment. The residual group is the part that differentiates the amino acids.

The ribosomes sequence amino acids according to DNA instructions forming the primary structure of the protein. To carry out its tasks, the protein evolves to higher level structures, by linking amino acids with hydrogen bonds (the secondary structure), folding the resulting polypeptide chain into three dimensional structures (the tertiary structure) and coupling together multiple chains (the quarternary structure). The tertiary structure of the protein is called the native state. Understanding how proteins fold into their native states is a major goal of bioinformatics.

A part of the protein folding problem is the protein structure prediction problem: devising good computational algorithms for prediction of the protein native state from its amino acid sequence. Designing such algorithms is an important research area of computational biology. The major contributions of computer based modeling for the PSP problem are acknowledged by the bioengineering community in the form of the CASP PSP competitions, which are held every two years since 1994.

Various simplified models for the protein structure exist (e.g. the Toy model, the Functional Model Protein — FMP, the Hydrophobic-Polar model — HP). Despite the simplicity of these models, the protein folding problem is still very hard. The high computational complexity of predicting the native state of a protein using information ab out the primary structure recommends this problem for metaheuristic approaches.

The HP Lattice Model

The full complexities of the folding processes that take place in proteins are yet to be unveiled. Even if they were known, detailed all atom simulations of such complex processes would not be possible for large proteins in modern computers. In order to tackle the protein folding problem, simpler models were developed.

In lattice models, amino acids can occupy fixed sites on a square lattice (other lattice types can also be considered [3]). The distances between adjacent amino acids are equal and the folding angles are multiple of 90 degrees. Lattice models are usually simple enough to be studied in detail, yet they include general principles which provide deep insights on the folding process.

The current theory on the causes of protein folding is that the hydrophobic interactions are the dominant component of the folding code [4]. With respect to this, the amino acids can be hydrophilic (water-attracting) or hydrophobic (water-rep elling). Hydrophilic amino acids (H) are electrically polarized, capable of bonding with Hydrogen, and soluble in water. Hydrophobic amino acids (P) are neutral from an electrical point of view, non-polar and they prefer other neutral non-polar solvents.

The HP model focuses only on the short-range contacts of hydrophobic amino acids and ignores any other type of interactions. The energy of the folding decreases for each pair of unbound hydrophobic amino acids which are next to each other in the lattice. Formally, the energy of a protein folding in the HP model it is computed by:

$$E = \sum_{i<j} \delta(p_i, p_j)$$

where p_i and p_j are the locations of hydrophobic amino acids *i* and *j*, and

$$\delta(p_i, p_j) = \begin{cases} 1, \text{ if } |i-j| > 1 \text{ and } \|p_i - p_j\|_1 = 1 \\ 0, \text{ otherwise} \end{cases}$$

where $\|\cdot\|_1$ denotes the Manhattan distance. Despite its simplicity, the HP model remains a focus of research in computational biology and statistical physics [5].

Even with very simple models, the protein folding problem is still very hard due to the large number of possible conformations and the many local minima in the energy landscape. In fact, the PSP problem in the HP model is a NP-hard problem [6].

A folded protein is a self avoiding chain of amino-acids. The most common folding encodings store the angles/directions of the bonds between amino acids. This approach ensures that the chaining hard constraint is satisfied (but not the self avoidance) and has low memory requirements. In absolute encoding, for each amino acid of the protein (except the first one) a folding code sets the next folding direction independently of the previous code. In relative encoding, for each amino acid (except the first two) a folding code sets the next folding direction in the context of the previous two codes.

Figure 1 presents the folding instructions on 2D square lattices for absolute and relative encodings. The relative encoding has less folding codes and instructions (i.e. smaller search space); the absolute encoding is more stable to changes.

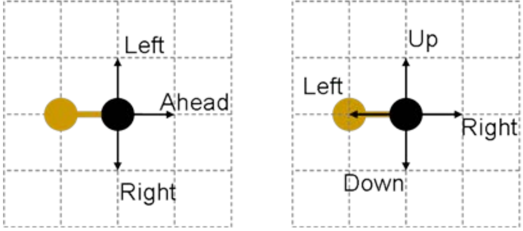

Figure 1. Absolute (left) and relative (right) folding instructions on a 2D lattice

PSO for Protein Structure Prediction

Particle Swarm Optimization is a meta-heuristic based on the principles of swarm intelligence, inspired by the social behavior of bird flocking. It uses a set of potential solutions (called particle swarm) to solve optimization problems. The objective function describes a specific optimization problem and defines the problem landscape in terms of solutions quality. Particles fly in the problem landscape on oscillating trajectories, searching for high quality solutions. The search process uses a simple inter-particle communication method that allows particles to develop a collab orative search b ehavior. For a thorough description on the PSO metaheuristic we refer the user to [7].

The real-valued PSO algorithm was applied successfully on the PSP problem with various off-lattice models [8–10]. Although the search space is larger, these off-lattice models give the PSO algorithm more freedom to explore it, more informative feedback, and provide a smooth energy landscape. The successes reported by real-valued PSO for off-lattice protein folding models are very encouraging. In previous papers [11-12] we have applied Particle Swarm Optimization (PSO) to the PSP problem with on-lattice models and found it to perform well compared to other meta-heuristic methods.

When working with lattice models, the number of possible codes for each folding operation depends on the dimensions of the lattice (i.e. 2D or 3D) and the folding representation (i.e. absolute or relative). In 2D lattices, there are 4 codes in absolute encoding (i.e.Up, Down, Left, Right) and 3 codes in relative encoding (i.e. Left, Right, Ahead). The search space of the PSP problem is the set of all valid protein conformations and the PSO algorithm needs to be able to access any point in it represented with these folding codes.

In our previous research [12], we used a version of the binary PSO [13] where groups of position elements (i.e. binary values) encode a single folding code. For the 2D HP lattice model, the minimum number of elements in the group is 2. In order to decode the protein folding represented by the position of a particle in the search space, each type of group values needs to be decoded into a valid folding instruction. Therefore, we used the following representation mappings:

- absolute encoding: Down = 00, Up = 01, Left = 10, Right = 11;

- relative encoding: Ahead = 00 and 01, Left = 10, Right = 11.

It should be noted that the Ahead code in relative encoding has two representations. This was a design decision which simplifies the implementation of the algorithm: a substitution procedure replaces 01 groups with 00 prior to evaluation.

If the protein contains n amino acids, then the position and velocity vectors of the particle have 2(n − 1) values for absolute encoding and, respectively, 2(n − 2) values for relative encodings.

Parameter optimization of PSO for Protein Structure Prediction

In our previous research [12] we used a particular protein as an initial test bench for selection adequate parameters for PSO and GAs in competing tests. Due to large number of possible configurations for PSO parameters, an exhaustive search of best parameters is impossible. Even if such a configuration would be found for a particular protein pool, is it very likely that it will not perform well on any given proteins, in accordance ot the No Free Lunch theorem.

Therefore, in this paper we research the use of genetic algorithms (GA) to automatically optimize the parameters of PSO for protein structure prediction for a given protein pool. To this end, a genetic algorithm [14] is used to search among the various parameter configuration for PSO the one that provides best result for the given protein pool.

The parameters for the GA we used are as follows:

- population size: 20 individuals
- runtime: 50 generations
- mutation probability: 10%
- crossover probability: 60%
- selection method: roulette wheel
- solution selection: best from all generations

The experimental setup is described next. For our experiments we used hybrid chromosomes for the genetic algorithm, which were encoded as binary strings (for ease of implementation). These hybrid chromosomes encode the major parameters of a PSO algorithm: individual learning factor, social learning factor, network topology, velocity limit and inertial factor.

For the individual and social learning factors, the GA will optimize the bounds of the liniar random number generator that PSO uses. The allowed values for their corresponding genes will encode numbers between 0.25 and 4 with a 0.25 precision scale. The 16 values allowed for each parameter are encoded in the GA chromosome using 4 bits.

For the network topology, the GA will select one of the standard PSO topologies: star, full, ring, and tree. These 4 topology types are encoded in the GA chromosome using 2 bits.

The GA selects values for the velocity limit from 0.5 to 4.5 with a 0.5 precision scale, using 3 bits to encode it in the chromosome.

For the inertia factor, GA allows for values between 0.5 and 1.5, with a 0,125 precision scale, encoded in a set of 3 bits.

Therefore, the total length of the GA chromosome is 16 bits. An exhaustive search of the best PSO performance within this parameter setup would require 65536 PSO experiment runs. Considering the computational effort required to run one PSO experiment for the PSP problem, an exhaustive search is much more demanding than a GA run with 20 individuals and 50 generations.

It is obvious that swarm size and number of algorithm iterations are missing from the previous list of PSO parameters that GA controls. Although they obviously affect the performance of the PSO algorithm, they also have an important role in the computational effort. Since we wanted to maintain control over the runtimes of the algorithm and to provide a common test ground for the GA chromosomes, we decided to fix these values to 100 particles and 100 iterations.

Also fixed in the PSO algorithm is the encoding method. We selected absolute encoding for ease of implementation.

For our experiments, we used the protein pool from [5]:

Test	Protein string	Length	Minimum energy
P1	3H 1P 1H 5P 1H	11	-2
P2	1H 1P 1H 2P 2H 1P 2H 1P 1H 1P 2H 2P 1H 1P 1H	20	-9
P3	2H 2P 1H 2P 1H 2P 1H 2P 1H 2P 1H 2P 1H 2P 2H	24	-9
P4	2P 1H 2P 2H 4P 2H 4P 2H	25	-8

	4P 2H		
P5	3P 2H 2P 2H 5P 7H 2P 2H 4P 2H 2P 1H 2P	36	-14

Table 1. The protein strings used in the protein pool test bench.

From the GA run, the individual with the best fitness over all iterations was selected as the solution of the algorithm. The genetic algorithm was run 20 times. The parameters identified by GA as producing good PSO setups for PSP are presented below.

For the individual and social learning factors, in 75% of cases the GA decided to use a value greater than 1, with an average of 1.65. We estimate that this value, which is larger than the one we used in previous experiments, produces more energy variation during the PSO run, which helps it to better explore the search space.

Regarding the network topology, the GA has selected one of the standard PSO topologies with the following frequencies: star - 42%, full - 36%, ring - 20%, and tree - 2%. It seems that GA favored the highly connected topologies (star and full). This can be explained by the speed of distribution of information throughout the networks. In the star and full network, the information about new low energy folds is quickly shared with the rest of the group, which can then focus on exploiting that particular information by attempting variations on that fold. It is interesting to notice, that star topology was the most preferred one. Our theory is that this happens because it provides a small amount of buffering, too, in contrast to the full topology. It would be interesting to test multi-level star (star of stars) topologies, but this is outside the scope of this article.

As expected, for the velocity limit, GA favored values larger than 3 in 85% of the cases. This large values allow the PSO particles to keep swarming around promising areas of the search space, trying to optimize the existing solutions.

For the inertia factor, GA selected values between 0.75 and 1.125 in 90% of the cases, which is in tune with the settings for this parameter recommended in the literature.

In all of the cases, PSO algorithm with the parameters settings found by GA was able to correctly identify the minimum energy configuration of the proteins test pool. This is in contrast with previous results in which the PSO algorithm found the minimum energy configurations only for the shorter proteins chains.

Conclusions

As indicated by our previous research, particle swarm optimization can be successfully used for the protein structure prediction problem. Furthermore, the its parameters can be optimized to improve its performance on given protein chains. However, the research presented in this paper could be extended further by providing better resolution to the values of the parameters and considering more complex configurations (like the star of stars topology). Ofcourse, this would require more computational resources, so a good balance between the optimization of PSO parameters and the actual PSO search process is needed, which is itself an optimization problem.

Bibliography

[1] Science: So Much More to Know . Science 309(5731) (2005) 78b–102

[2] Dill, K.A., Ozkan, S.B., Weikl, T.R., Chodera, J.D., Voelz, V.A.: The protein folding problem: when will it be solved? Current Opinion in Structural Biology 17(3) (2007) 342 – 346 Nucleic acids / Sequences and topology.

[3] Bockenhauer, H.J., Ullah, A.Z.M.D., Kapsokalivas, L., Steinhofel, K. In: A Local Move Set for Protein Folding in Triangular Lattice Models. Springer (2008) 369–381

[4] Dill, K.: Dominant forces in protein folding. Biochemistry 31(29) (1990) 7133–7155

[5] Santana, R., Larranaga, P., Lozano, J.A.: Protein folding in simplified models with estimation of distribution algorithms. IEEE Transactions on Evolutionary Computation 12(4) (2008) 418 – 438

[6] Berger, B., Leighton, T.: Protein folding in the hydrophobic-hydrophilic (HP) is NP-complete. In: Proc. of RECOMB ’98, New York, NY, USA, ACM (1998) 30–39

[7] Clerc, M.: Particle Swarm Optimization. HERMES Science Publishing Ltd, London (April 2006)

[8] Liu, J., Wang, L., He, L., Shi, F. In: Analysis of Toy Model for Protein Folding Based on Particle Swarm Optimization Algorithm. Volume 3612. Springer Berlin/Heidelberg (2005) 636–645

[9] Perez-Hernandez, L.G., Rodrıguez-Vazquez, K., Garduno-Juarez, R.: Parallel particle swarm optimization applied to the protein folding problem. In: Proc. Of GECCO ’09, New York, NY, USA, ACM (2009) 1791–1792

[10] Zhu, H., Pu, C., Lin, X., Gu, J., Zhang, S., Su, M.: Protein structure prediction with EPSO in Toy model. In: Proc. of ICINIS ’09, IEEE Press (2009) 673–676

[11] Bautu A, Generalizations of Particle Swarm Optimization: Applications of Particle Swarm algorithms to Statistical Physics and Bioinformatics problems, LAP LAMBERT Academic Publishing, ISBN 978-3848417315

[12] Băutu, A, Luchian, H, Protein Structure Prediction in Lattice Models with Particle Swarm Optimization, In Swarm Intelligence, Springer Berlin Heidelberg, ISBN 978-3-642-15460-7

[13] Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: Proc. of 1997 Conference on Systems, Man, and Cybernetics. Volume 5., Piscataway NJ, USA, IEEE Press (Octomber 1997) 4104–4109

[14] Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, New York (1996)