Lopes HS, Bitello R. A differential evolution approach for protein folding using a lattice model. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 22(6): 904~908 Nov. 2007

A Differential Evolution Approach for Protein Folding Using a Lattice Model

Heitor Silverio Lopes and Reginaldo Bitello

Bioinformatics Laboratory, Federal University of Technology – Paraná, Av. 7 de setembro, 3165 80230-901 Curitiba, Brazil

E-mail: hslopes@pesquisador.cnpq.br; regi.bitello@gmail.com

Received November 22, 2006; revised May 22, 2007.

Abstract Protein folding is a relevant computational problem in Bioinformatics, for which many heuristic algorithms have been proposed. This work presents a methodology for the application of differential evolution (DE) to the problem of protein folding, using the bi-dimensional hydrophobic-polar model. DE is a relatively recent evolutionary algorithm, and has been used successfully in several engineering optimization problems, usually with continuous variables. We introduce the concept of genotype-phenotype mapping in DE in order to provide a mapping between the real-valued vector and an actual folding. The methodology is detailed and several experiments with benchmarks are done. We compared the results with other similar implementations. The proposed DE has shown to be competitive, statistically consistent and very promising.

Keywords bioinformatics, differential evolution, evolutionary computation, protein folding

1 Introduction

Proteins are composed by amino acids chains where there is all the information necessary for generating a unique tri-dimensional structure. The exact way proteins fold just after being synthesized in the ribosome is unknown. As consequence, many computational approaches of different levels of complexity have been proposed to simulate the folding of proteins. However, to date, even simple models are still computationally expensive. Recently, several methods have been proposed in the quest of solving the protein folding problem (PFP), such as Monte Carlo simulation[1], genetic algorithms[2] and ant colony optimization[3]. Despite being an important issue in bioinformatics, there is still no efficient method for solving the PFP.

The objective of this work is to evaluate the applicability of the Differential Evolution algorithm to the PFP using the 2D-HP model, and to compare its performance with other similar algorithms recently published.

2 2D-HP Model

Amongst the several discrete models used to simulate how a protein folds, the hydrophobic-polar (HP) is, possibly, the most simple and most widely studied model. The HP model was proposed by Dill[4], who demonstrated that some behavioral properties of real-world proteins could be inferred by using this simple model. In this model, the amino acids of a protein are considered either hydrophobic (aversion to water) or polar (affinity to water, the same as hydrophilic). Despite the simplicity of this lattice model, exact algorithms to solve the problem were proved to be NP-hard[5].

Fig.1 shows the 2D-HP model for a hypothetical

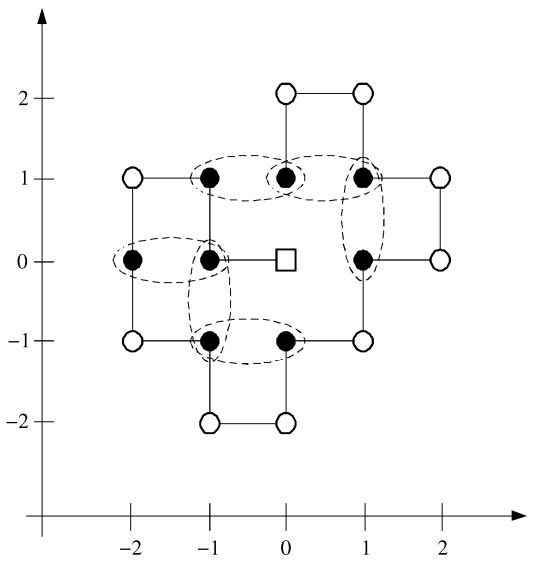

Fig.1. Example of H-H contacts in the 2D-HP model.

Short Paper

This work is supported by the Brazilian National Research Council under Grant No.305720/04-0.

18 amino acids-long protein, folded in such a way that 6 non-local H-H contacts occur. In this figure, black and white dots are, respectively, hydrophobic and polar amino acids. The square dot is the first element of the chain, and H-H contacts are represented by dotted lines.

The free energy function of a conformation, suggested by [6], is represented in (1):

$$E = \sum_{i<j} e_{r_i r_j} \Delta(r_i - r_j) \tag{1}$$

where: $\Delta(r_i - r_j) = 1$ if amino acids r_i and r_j are not consecutive in the chain, and $\Delta(r_i - r_j) = 0$, otherwise. Depending on the type of contacts between amino acids r_i and r_j, the energy $e_{r_i r_j}$ will be e_{HH}, e_{HP} or e_{PP}, corresponding to H-H, H-P and P-P contacts, respectively. According to [6] this model satisfies the following physical limitations:

1) Compact conformations have a smaller energy value than any other non-compact conformation.

2) Hydrophobic amino acids will be buried inside the conformation, as most as possible. This idea is expressed by the relationship $e_{\mathrm{PP}} > e_{\mathrm{HP}} > e_{\mathrm{HH}}$, that decreases the energy of conformations in which the Hs are hidden inside.

3) Different types of amino acids tend to get apart. This is expressed by the relationship: $2e_{\mathrm{HP}} > e_{\mathrm{PP}} > e_{\mathrm{HH}}$.

3 Methodology

3.1 Differential Evolution

Differential Evolution (DE) is an evolutionary computation method invented by Storn and Price[7] for numerical optimization. The central idea of this algorithm is the use of difference vectors for generating perturbations in a population of vectors. This algorithm is conceptually simple and, at most times, converges fast to a good solution. Besides, DE is robust and has few parameters do be tuned. Consequently, it has drawn attention of researchers who have studied its utility for complex optimization problems[8,9]. Fig.2 shows graphically how the vector operations take place in a 2-dimensional space, at a given generation. A detailed description of DE can be found in [10].

3.2 Vector Encoding

In DE, the variables of the problem are encoded in a vector and, usually, the meaning of its elements to the real-world is straightforward. Consequently, the concept of genotype, as in genetic algorithms, is not applicable in the original DE. However, for the specific problem dealt in this work, the adaptation devised to represent possible solutions to the PFP in a real-valued vector requested the establishment of a genotype-phenotype mapping. Individuals in DE are real-valued vectors which, in turn, are decoded into a specific fold of an amino acid chain in a square lattice. The reason for this approach was to use the original DE algorithm, without significant changes.

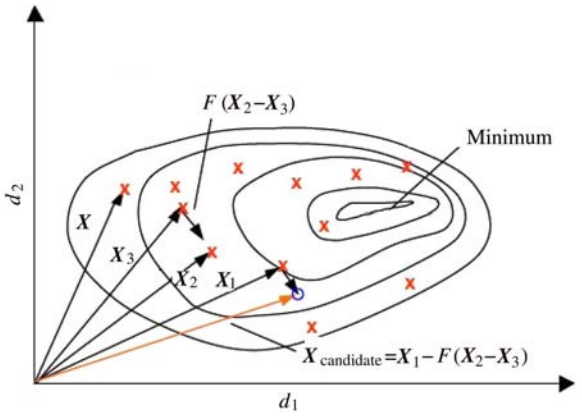

Fig.2. Representation of a candidate vector, obtained by means of vector operations.

Basically, there are three ways of representing an amino acid chain in a lattice using the HP model: Cartesian coordinates, internal coordinates and geometrical distances. The proposed implementation with DE uses relative internal coordinates. This coordinates system implicitly assures that the connectivity of the amino acids chains will be preserved when a given conformation is drawn in the lattice. This property avoids loosing time checking the validity of a given conformation.

Using the relative internal coordinates system in a bi-dimensional space, there are three possible movements: (F)orward, (R)ight and (L)eft. Therefore, the phenotypical representation of a solution is defined over the alphabet of movements $\{F, R, L\}$. A given folding of N amino acids, represented by an N-dimensional vector, is defined by a string with $N-1$ movements. The genotypical representation is the usual real-valued vector of a regular DE. Considering x_{ij} the j-th element of vector $\boldsymbol{X}_i$, P the string representing the sequence of movements of the folding, and $\alpha < \beta < \delta < \gamma$ arbitrary constants in $\mathcal{R}$, the genotype-phenotype mapping is defined as follows:

$$\begin{cases} \text{If } \alpha < x_{ij} \leqslant \beta \text{ then } P_j = L, \\ \text{If } \beta < x_{ij} < \delta \text{ then } P_j = F, \\ \text{If } \delta \leqslant x_{ij} < \gamma \text{ then } P_j = R. \end{cases} \tag{2}$$

Notice that the proposed mapping allows to privilege some movements by enlarging the corresponding range in which it is defined (or narrowing the other

ranges). This strategy can be useful during evolution considering the characteristics of a specific folding. Furthermore, this mapping allows several genotypes to represent a single phenotype.

The proposed encoding is also valid for a 3D model, where there are two additional movements, relative to the 2D model: (U)p and (D)own. The same strategy for genotype-phenotype encoding defined before could be used, but with additional constants to define the range of the (U)p and (D)own movements.

3.3 Constraint Handling

When applying DE to a constrained problem, such as the PFP, unfeasible solutions may appear during evolution. There are three basic strategies for dealing with an unfeasible solution: discard it, fix it, or accept it with a penalty proportional to the extent of violations of the constraints. The last alternative is especially interesting when there is some chance of the violations being fixed by themselves along the evolution. For simplicity, we adopted the first alternative: when an individual represents an invalid folding (that is, there is more than one amino acid in a given position in the lattice), it is simply discarded. Further work will evaluate the other strategies.

3.4 Initial Population

The simplest way to generate the initial population without invalid individuals at phenotypical level is creating "stretched" individuals, that is, the string is composed only of F's. Although the initial population is exactly the same at the phenotypical level, all their elements are quite different at genotypical level, thanks to the genotype-phenotype mapping defined before. This procedure guarantees an initial population with reasonable diversity of valid individuals, a necessary condition for evolving good solutions.

3.5 Decoding and Fitness Function

Before evaluating an individual, the real-valued vector is decoded into a string based on the alphabet $\{F, R, L\}$. In the example of Fig.1 for chain "*PHHPHPHPPPHPHPPHPPPH*", the corresponding string of relative movements is "*LRLLFLRLLRLRLLRLL*". Next, this folding is evaluated by counting the number of non-local H-H contacts, as defined in Section 2. This fitness is based on the assumption that the non-local H-H contacts are the main force driving the folding of a protein. Therefore, the fitness function is aimed at maximizing of the number of non-local H-H contacts.

3.6 Strategies

Storn and Price[10] developed a set of strategies that allow a large number of options, depending on the nature of the problem. Such strategies are classified as follows.

1) Vector to be disturbed: it can be a randomly chosen vector of the population (***rand***) or the vector with the best fitness value (***best***). Vectors randomly chosen lead to a richer diversity, whereas using the other option, the convergence will be faster.

2) Number of weighted differences: for a small population the weighted difference of only two vectors is more usual. For larger populations authors[10] have shown that four vectors are more effective regarding convergence.

3) Crossover type: it can be binomial (*bin*), when all the elements of the vector have the same probability CR; and exponential (*exp*), when crossover is done whenever a randomly chosen value is less or equal to CR.

The choice of the strategy is done by trial-and-error, since there is still no well-established procedure for choosing the best strategy for a given problem.

An interesting approach for keeping the diversity of the population along the search, but at the same time facilitating convergence, is alternating strategies, as follows. Use the strategy $Best2Exp$[10] while some improvement is observed in the best fitness for the last N generations. This strategy aims at a fast convergence. Next, when the number of generations without improvement in the best fitness is equal to or larger than N, change to strategy $Rand2Exp$, and keep it for up to M generations without improvement. This last strategy aims at improving diversity. Case the best fitness is improved or if M generations without improvement were done, turn back to the $Best2Exp$ strategy, clear counters N and M, and repeat the cycle.

4 Experiments and Results

For testing the DE algorithm, we used a benchmark of 9 synthetic amino acids chains found in the literature[2,3,11,12], ranging from 20 to 85 amino acids. Table 1 shows the instances used, including the number of amino acids, the amino acids chain translated to the HP model, and the maximum known number of non-local H-H contacts.

Due to the stochastic nature of DE, for each test instance, 100 runs were done, using different random seeds. Results reported are the average values over these 100 runs.

For all experiments, the following parameters were used:

population size = number of amino acids $\times$ 15;

crossover probability $CR = 80\%$;
weighting factor $F = 0.85$.

Also, we used the alternating strategies *Best2Exp* and *Rand2Exp*, as explained before, with $N = 100$ and $M = 70$. The constants that define the ranges for mapping the genotype to the phenotype were: $\alpha = -3$, $\beta = -1$, $\delta = +1$, $\gamma = +3$. The software was developed in C programming language and all experiments were run in a PC with Athlon X2 64 bits processor with 1 Gbytes RAM.

Table 1. Benchmarks Used in the Experiments

n	HP Chain	E
20	$HPHP^2H^2PHP^2HPH^2P^2HPH$	9
24	$H^2P^2HP^2HP^2HP^2HP^2HP^2HP^2H^2$	9
25	$P^2HP^2H^2P^4H^2P^4H^2P^4H^2$	8
36	$P^3H^2P^2H^2P^5H^7P^2H^2P^4H^2P^2HP^2$	14
48	$P^2HP^2H^2P^2H^2P^5H^{10}P^6H^2P^2$ $H^2P^2HP^2H^5$	23
50	$H^2PHPHPHP^4HP^3HP^3HP^4$ $HP^3HP^3HPH^4PHPHPHPH^2$	21
60	$P^2H^3PH^8P^3H^{10}PHP^3H^{12}P^4$ H^6PH^2PHP	36
64	$H^{12}PHPHP^2H^2P^2H^2P^2HP^2H^2P^2$ $H^2P^2HP^2H^2P^2H^2P^2HPHPH^{12}$	42
85	$HPHPH^{16}P^4H^{12}P^6H^{12}P^3H^{12}P^3$ $H^{12}P^3HP^2H^2P^2H^2P^2HPH$	52

Table 2. Comparison of Results Using Different Approaches

n	E	[1]	[3]	[2]		DE	
				max	avg	max	avg
20	9	9	9	9(74)	8.74	9(100)	9.00
24	9	9	9			9(100)	9.00
25	8	8	8			8(100)	8.00
36	14	14	14	14(6)	12.44	14(96)	13.96
48	23	23	23	23(2)	20.06	23(100)	23.00
50	21	21	21			21(100)	21.00
60	36	36	36			35(79)	34.79
64	42	42	42	40(1)	33.58	42(88)	41.87
85	53	52	53	51(2)	45.74	52(50)	51.38

Table 2 presents the results obtained by our approach and the comparison with others. In this table, the first column shows the number of amino acids of the instance. The second column shows the maximum number of non-local H-H contacts known to date. The next two columns are the best results found by PERM, a Monte-Carlo-based algorithm[1], and by an Ant Colony Optimization algorithm[3], respectively. The fifth and sixth columns are the results obtained by using a genetic algorithm with enhanced operators[2]: first the maximum number of non-local H-H contacts found by the algorithm and, within parenthesis, the number of times this maximum was found in 100 independent runs, and, next, the number of non-local H-H contacts averaged over 100 runs. The last two columns show the results obtained by the proposed DE algorithm described in this paper. The meaning of these columns are the same as the fifth and sixth columns.

For sequences up 50 amino acids long, our proposed algorithm took few seconds per run. Sequences with 60 and 64 amino acids took an average of 108 and 1206 seconds per run. For the largest sequence, our algorithm needed around 10 000 seconds each run.

5 Discussion and Conclusions

For chains up to 50 amino acids, all algorithms have found the maximum number of non-local H-H contacts. However, our DE approach was much more consistent than the GA algorithm, since it achieved the maximum in almost all runs of all chains. Only for the 36 amino acids-long chain, DE failed to find the maximum in 4 out of 100 runs. There are no information for PERM and ACO to compare with our approach, regarding this issue. For the chain with 64 amino acids, all algorithms achieved the maximum, but DE performed much better than the GA, regarding any evaluation parameter. For chains with 60 and 85 amino acids, our DE approach did not achieve the maximum, when compared with the ACO, but PERM did not too for the 85 amino acids chain. However, it is remarkable the consistency of the proposed algorithm when observing not only the average, but also, the number of times the maximum was found for all instances. This fact is very important for a stochastic algorithm, and suggests that DE has a better repeatability than the GA.

We have proposed a methodology for using the differential evolution algorithm for the protein folding problem with the 2D-HP model. The DE algorithm was kept as originally described by [7], and we introduced the concept of genotypical-phenotypical mapping. Thanks to this mapping, the DE algorithm, originally devised for real-valued vectors, could be used for evolving solutions to the PFP. Considering that the selection method used is based only in the fitness function (which is based on the phenotypical representation), it is possible that promising individuals (seen at the genotypical level) could be discarded along generations. Other implications of the proposed genotypical-phenotypical mapping are still under study and will be focused in future work.

It is important to note that no serious attempt was done to optimize parameters of the algorithm, neither to adjust the range of constants defined in (2). As a consequence, it is fair to expect that even better results (than those shown in Table 2) could be achieved, or, at least, similar results could be achieved with smaller

computational effort. Besides, we re-emphasize that we used the basic DE, while the other results cited were obtained with much more elaborated and improved versions of the algorithms (PERM, ACO and GA). To avoid parameter adjustments without prior knowledge of the behavior of the algorithm for this specific problem or specific instances, future work will focus on a self-adaptive DE, allowing the own algorithm to adjust the values of its parameters during evolution[13].

As the length of the amino acids chain increases, the problem gets harder. In fact, the lattice model and the energy function (see (2)), based only on the number of non-local H-H contacts, leads to a strongly multimodal fitness landscape with many equal-sized plateaus. This fact, by itself, makes the problem even harder for any stochastic heuristic method. Even so, the DE approach seems to be very promising.

Protein folding using the 2D-HP model is an important, and still opened problem, in bioinformatics, since the efficiency of methods is bounded by the length of sequences. We believe that the proposed algorithm is an innovative and useful contribution to this area of research, because it is competitive, consistent and promising. Future work will focus on self-adapting DE parameters and using this approach for dealing with more complex models of protein folding, such as encoding the internal torsion angles.

References

[1] Chikenji G, Kikuchi M, Iba Y. Multi-self-overlap ensemble for protein folding: Ground state search and thermodynamics. *Physical Review Letters,* 1999, 83(9): 1886~1889.

[2] Lopes H S, Scapin M P. An enhanced genetic algorithm for protein structure prediction using the 2D hydrophobic-polar model. *Lecture Notes in Computer Science,* 2005, 3871: 238~246.

[3] Shmygelska A, Hoos H H. An improved ant colony optimisation algorithm for the 2D HP protein folding problem. *Lecture Notes in Computer Science,* 2003, 2671: 400~417.

[4] Dill K A. Theory for the folding and stability of globular proteins. *Biochemistry,* 1985, 24(6): 1501~1509.

[5] Crescenzi P, Goldman D, Papadimitriou C *et al.* On the complexity of protein folding. *Journal of Computational Biology,* 1998, 5(3): 423~465.

[6] Li H, Helling R, Tang C *et al.* Emergence of preferred structures in a simple model of protein folding. *Science,* 1996, 273(5275): 666~669.

[7] Storn R M, Price K V. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization,* 1997, 11(4): 341~359.

[8] Coelho L S, Lopes H S. Supply chain optimization using chaotic differential evolution method. In *Proc. IEEE Systems, Man and Cybernetics Conference*, Piscataway, NJ, 2006, IEEE Press, pp.3114~3119.

[9] Yang J, Wongsa S, Kadirkmanathan V *et al.* Differential evolution and its application to metabolic flux analysis. *Lecture Notes in Computer Science,* 2005, 3449: 115~124.

[10] Price K V, Storn R M, Lampinen J A. Differential Evolution – A Practical Approach to Global Optimization. Berlin: Springer-Verlag, 2005.

[11] Konig R, Dandekar T. Improving genetic algorithms for protein folding simulations by systematic crossover. *Biosystems,* 1999, 50(1): 17~25.

[12] Unger R, Moult J. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology,* 1993, 231(1): 75~81.

[13] Maruo M H, Lopes H S, Delgado M R B S. Self-adapting evolutionary parameters: Encoding aspects for combinatorial optimization problems. *Lecture Notes in Computer Science,* 2005, 3448: 154~165.

Heitor Silverio Lopes obtained the B.Sc. and M.Sc. degrees in electrical engineering and computer science from the Federal University of Technology of Parana (UTFPR), Brazil, respectively in 1979 and 1990, and a Ph.D. degree from the Federal University of Santa Catarina, in 1996. Currently, he is an associate professor in the Department of Electronics and head of the Laboratory of Bioinformatics of UTFPR. Dr. Lopes has served as member of program committee of many international conferences and editorial board of Journal of Computational Intelligence and Bioinformatics. To date, Dr. Lopes has supervised 25 M.Sc. dissertations and Ph.D. theses, and published more than 150 papers in conferences and scientific journals. His current research interests are: evolutionary computation, bioinformatics and reconfigurable computing.

Reginaldo Bitello graduated in computer science by the Foundation for Social Studies of Parana, Brazil. He received an M.Sc. degree in computer science from the Federal University of Technology of Parana in 2007. He was a specialized technician to the Brazilian air force and he is currently working for the Parana state government as computer systems analyst.