Hierarchical Parallel Genetic Algorithm applied to the three-dimensional HP Side-Chain Protein Folding Problem

César Manuel Vargas Benítez, Heitor Silvério Lopes
Bioinformatics Laboratory, Federal University of Technology – Paraná (UTFPR),
Av. 7 de setembro, 3165 80230-901, Curitiba (PR), Brazil
cesarvargasb@gmail.com, hslopes@utfpr.edu.br

***Abstract*—This work describes a Hierarchical Parallel Genetic Algorithm (HPGA) applied to the Protein Folding Problem (PFP). The modeling of the problem, using the 3DHP-Side-chain model, and details of the HPGA are presented. The effect of the energy weights in the performance of the algorithm was also studied. The HPGA was tested using three sets of benchmark sequences. Results show that the HPGA obtained biologically coherent results, suggesting the adequacy and efficiency of the HPGA for the problem.**

***Index Terms*—Genetic Algorithm, Bioinformatics, Protein Folding, 3DHP-SC.**

I. Introduction

Proteins are essential to life and they have countless biological functions. They can be defined as polymers composed by a chain of amino acids (also known as residues) that are linked together by means of peptide bonds. Each amino acid is characterized by a central carbon atom (referred as Cα), to which a hydrogen atom, an amine group (NH_2), a carboxyl group (COOH) and a side-chain (also known as radical R) are attached. The carboxyl group of one amino acid and the amino group of another one are responsible for linking them together. An amino acid differs from others by their side-chain, since the backbone of all amino acids are the same [18].

Proteins are synthesized in the ribosome of cells following a template given by the messenger RNA (mRNA). During the synthesis, the protein folds itself into a unique three-dimensional structure. This process is known as protein folding. The specific shape to which the protein naturally folds is known as its native conformation. The biological function of a protein depends on its three-dimensional shape, which, in turn, is a function of its primary structure (linear sequence of amino acids). It is known that ill-formed protein (due to wrong folding) can be completely inactive or even harmful to the organism. Several diseases are believed to result from the accumulation of ill-formed proteins. Therefore, better understanding the protein folding process, the three-dimensional structure and functionality of proteins, is a fundamental issue for Medicine/Biochemistry. Notwithstanding, despite a large number of proteins that have been discovered by recent genome sequencing projects, only a small amount of these proteins have their three-dimensional structure kwnown. For instance, the UniProtKB/TrEMBL [8] repository of protein sequences has currently around 11.2 million records (as in July/2010), and the Protein Data Bank – PDB [5] has the structure of only 62,212 proteins [1]. This fact is due to the cost and difficulty in unveiling the structure of proteins, from the biochemical and biological point of view. It is here that Computer Science has an important role, proposing models for studying the Protein Folding Prediction (PFP) problem [16].

The simplest computational model for the PFP problem is known as Hydrophobic-Polar (HP) model, both in two (2D-HP) and three (3D-HP) dimensions [9]. However, the computational approach for searching a solution for the PFP using simple HP models was proven to be NP-complete [2]. Consequently, metaheuristic approaches seem to be the most reasonable algorithmic choice for dealing with the problem. Evolutionary computation methods and, in special, Genetic Algorithms (GA) have been proved not only adequate, but very efficient for the PFP [16], [20]

The objective of this work is to extend and deepen a previous work [3], [4], aiming at finding the native conformation of synthetic proteins represented with the 3D-HP Side-Chain model. This is accomplished by using a hierarchical parallel genetic algorithm. We also studied the effect of the energy weights in the performance of the algorithm, proposing optimized values, not yet available in the literature for this model. Comparing with our previous work [3], [4], this version also includes a strategy for improving performance of the GA, updated results and new benchmarks.

II. The 3DHP Side-Chain Model

The Hydrophobic-Polar (HP) model is the most simple abstraction of the protein structure and divides the 20 standardized amino acids into two types: Hydrophilic (or Polar) and Hydrophobic. Therefore, a protein (string of amino acids) is represented by a string of characters defined over a binary alphabet $\{H, P\}$. The HP is a lattice model, and thus the chain is embedded in a square (for the 2D-HP) or cubic (for the 3D-HP) lattice. Both 2D-HP and 3D-HP models have been extensively explored in the recent literature [16].

From the biological point of view, the expressiveness of the HP models is very poor. Therefore, the next step to simulate

[1] Available, respectively, at http://www.uniprot.org/ and http://www.pdb.org

978-1-4244-6588-0/10/$25.00 ©2010 IEEE

more realistic features of proteins is to include a side bead representing the side-chain (SC) of the amino acids [15]. Recall that all the standard amino acids have the same basic structure (backbone), but different side-chains define their physico-chemical properties. Therefore, a protein is modeled by a common backbone and a side-chain, either Hydrophobic (H) or Polar (P). Different from the 2D-HP and 3D-HP, the 3D-HP-SC model is very sparsely studied in the recent literature, possibly due to the higher level of complexity of the model, when compared with the former. In fact, this model increases the realism of the simulation, but at the expense of increasing the complexity of the problem, from the computational point of view.

In the original HP model, it is considered that interactions between hydrophobic amino acids represent the most important contribution to the free-energy of the protein. The more hydrophobic interactions, the small the free-energy of the protein. For the 3D-HP-SC model, the free-energy of a given conformation is also in accordance with that principle, and takes into account the position in the space of the side-chains. To compute the energy of a conformation, Li et al. [15] proposed an equation that considers only three types of interactions (not making difference between types of side-chains). In this work we propose a more realistic way to compute the energy of a folding that accounts for all possible types of interactions, as shown in Equation 1.

$$\begin{aligned} H = \epsilon_{HH} \cdot \sum_{i=1,j>i}^{n} \delta_{r_{ij}^{HH}} + \epsilon_{BB} \cdot \sum_{i=1,j>i+1}^{n} \delta_{r_{ij}^{BB}} \\ + \epsilon_{BH} \cdot \sum_{i=1,j\neq i}^{n} \delta_{r_{ij}^{BH}} + \epsilon_{BP} \cdot \sum_{i=1,j\neq i}^{n} \delta_{r_{ij}^{BP}} \\ + \epsilon_{HP} \cdot \sum_{i=1,j>i}^{n} \delta_{r_{ij}^{HP}} + \epsilon_{PP} \cdot \sum_{i=1,j>i}^{n} \delta_{r_{ij}^{PP}} \qquad (1) \end{aligned}$$

In this equation, ϵ_{HH}, ϵ_{BB}, ϵ_{BH}, ϵ_{BP}, ϵ_{HP}, ϵ_{PP} are the weights of the energy for each type of interaction, respectively: hydrophobic side-chains (HH), backbone-backbone (BB), backbone-hydrophobic side-chain (BH), backbone-polar side-chain (PH), hydrophobic-polar side-chains (HP), and polar side-chains (PP). In a chain of n residues, the distance (in the three-dimensional space) between the i^{th} and j^{th} residue interacting with each other is represented by r_{ij}^{**}. For the sake of simplification, in this work we used unity distance between residues ($r_{ij}^{**} = 1$). Therefore, δ is an operator that returns 1 when the distance between the i^{th} and j^{th} side-chain is the unity, or 0 otherwise.

In section IV-C an optimized set of weights will be proposed for using in the experiments. Such set was found by means of a Genetic Algorithm.

As the amino acids chain folds over themselves, bonds (or contacts) between them take place, according to the possible interactions indicated in Eq. 1. It is believed that the non-local hydrophobic interactions are the main driving force that causes the macromolecule to fold correctly. During the folding process, the free energy of the protein tends to decrease. As mentioned before, the free-energy of a given three-dimensional conformation is inversely proportional to the number of non-local hydrophobic side-chain bonds (HnC). Therefore, an algorithmic procedure for the protein folding that maximizes the HnC will, probably, take the molecule to the smallest possible free-energy state.

According to [15], the weight for HnC (ϵ_{HH}) is negative. Consequently, the smaller the value of the free-energy function, the closer to its native state the conformation will be, in accordance with the Anfinsen's thermodynamic hypothesis [1]. In this work we consider the symmetric of H to turn the problem to maximization, and make the interpretation easier.

III. Hierarchical Parallel Genetic Algorithm for the PFP

Genetic Algorithms (GAs) are based on the Darwinian model of natural selection and evolution and they have been applied successfully to a wide range of problems. GAs operate on a population of individuals, as potential solutions to a given problem. Variables of a problem are represented by genes encoded in the individualt's chromosome, typically a string (or another alphabet) of binary digits.

In general, GAs are able to find good solutions in reasonable amount of time, but as they are applied to larger and harder problems, significant increment of processing time are required to find satisfactory solutions. As a consequence, there have been multiple efforts to make GAs faster, and one of the most promising options is to use parallel implementations. Besides, depending on the parallelization model, one can take advantage of the co-evolution between populations that may lead to better solutions.

Three main types of Parallel GAs (PGAs) [7] can be identified: global single-population master-slave, single-population fine-grained and multiple-population coarse-grained. There is also a combination of these types, leading to Hierarchical Parallel Genetic Algorithms (HPGAs). The HPGA implemented in this work has two levels: in the upper level multiple-population coarse-grained islands, and in the lower level global single-population master-slaves. This combination aims at taking advantage of the benefits of both approaches, as suggested by [6].

In the lower level the processing load is divided into several slaves, under the coordination of a master which is responsible for initializing the population, executing the selection procedure, applying the genetic operators, and distributing individuals to slaves. Slaves, in turn, receive a number of individuals, decode the corresponding chromosomes, compute penalties and the objective function. Finally, they return to the master the fitness value for each individual.

In the upper level each population (master and corresponding slaves) is seen as an island. Sporadic migrations take place between islands, controlled by four parameters set by the user: Migration gap (number of generations between successive migrations), Migration rate (number of individuals that will migrate at each migration event), Selection/Substitution criteria for migrants, and topology of connectivity between islands.

A. Encoding and Initial Population

The way variables are encoded has a significant influence in the dynamics and efficiency of a GA. The genotype is the way information is encoded in the chromosomes, and the phenotype is the decoding of such information into a real-world solution. To model the PFP, the phenotype represents the spatial position of the amino acids in a lattice. There are several ways for representing a folding in a chromosome [16], such as: distance matrix, Cartesian coordinates or internal coordinates. Most works in recent literature use this last approach, since it was suggested that it is the most efficient when using GA [14]. In this coordinate system, a given conformation is represented by a set of movements of one amino acid relative to its predecessor in the chain. Therefore, for a protein with n amino acids, a folding encoded in the chromosome of the GA will have $n - 1$ elements.

There are five possible movements for the backbone in the 3D space: (**L**eft, **F**ront, **R**ight, **D**own, **U**p), and the same number for the side-chains, (**l**eft, **f**ront, **r**ight, **d**own, **u**p). Combining all possible movements, there are 25 possibilities for each amino acid: {Ll, Lf, Lr, Ld, Lu, Fl, Ff, Fr, Fd, Fu, Rl, Rf, Rr, Rd, Ru, Dl, Df, Dr, Dd, Du, Ul, Uf, Ur, Ud, Uu}. Each element of this set is translated into an unique symbol. Instead of the traditional binary alphabet, a set of 25 numbers and letters was used to encode the chromosome (see Table I) [3].

Table I
ENCODING SCHEME OF THE RELATIVE INTERNAL COORDINATES.

Movements		Backbone				
		L	**F**	**R**	**D**	**U**
Side-chain	**l**	0	5	A	F	K
	f	1	6	B	G	L
	r	2	7	C	H	M
	d	3	8	D	I	N
	u	4	9	E	J	O

To represent the position of the amino acids in the cubic lattice, the Cartesian coordinates of each element (backbone and side-chain) will be later defined by a vector (x_i, y_i, z_i). This vector is obtained from the relative movement of an amino acid and position of its predecessor. A folding begins in the origin of the three-dimensional Cartesian coordinates, such that the first backbone is at $(0, 0, 0)$ and its side-chain at $(0, -1, 0)$. The position of the remaining amino acids is computed following the movements encoded in the chromosome. Therefore, a progressive sequential procedure is necessary for genotype-phenotype decoding. Figure 1 shows an example of genotype-phenotype decoding. Only the first four movements are shown in the figure due space restrictions.

Despite the advantages of using the proposed genotypical representation, it allows that two or more elements (backbone or side-chain) to occupy the same position in the lattice. This fact is known as collision, and results in an invalid conformation, since it is physically unfeasible. However, when the initial population is randomly created, the number of collisions tend to increase as the size of the protein increases [16]. Therefore, the GA will spend a reasonably large time throughout the first generations working with invalid individuals until good individuals appear. Aiming at improving the performance of the GA, a method for creating better initial individuals was proposed. The initial population is divided into two parts (80% and 20%). The first part is randomly generated, as usual, and the second part is composed only by collision-free individuals, generated by a backtracking strategy, explained below.

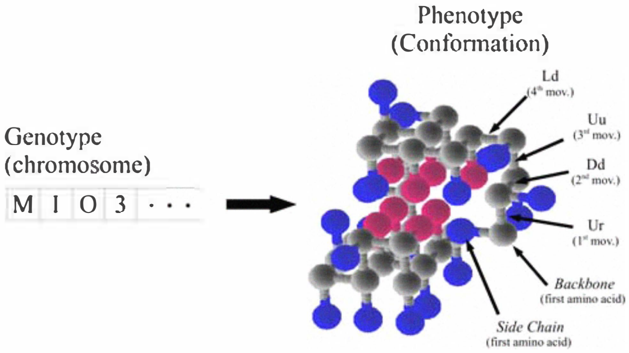


Figure 1. Example of genotype-phenotype decoding

After positioning the first backbone and its side-chain in the lattice, the movement of the next amino acid backbone is randomly selected. If the movement leads to a collision with the backbone or the side-chain of any other amino acid previously positioned in the lattice, a backtracking is done. Other possible positions for the backbone and side-chain are examined until a suitable combination is found (with no collisions). If this is not possible, the last pair backbone/side-chain is removed from the current position of the lattice and set to another position. The procedure is recursively repeated until a complete valid folding is obtained.

Although the proposed method for generating the initial population is very time-consuming, it assures the quality of individuals in the first generation, thus fostering the evolution of the AG towards good solutions.

B. Fitness Function

The objective function used in this work was first proposed by [17], and adapted to the 3DHP-SC by [3]. In a simplified way, this function has three terms (as shown in Equation 2). The first one is relative to the free-energy of the folding (see Equation 1), decreased by the number of collisions in the lattice. The following terms represent the compacity of the hydrophobic and polar side-chains, respectively. This is accomplished by means of the computation of the radius of gyration of the corresponding side-chains. This is done in such a way to favor conformations in which hydrophobic side-chains are compacted within the core, and polar side-chains are pushed outwards of the conformation. A detailed description of the objective function can be found in [3],[17].

$$fitness = Energy \cdot RadiusG_H \cdot RadiusG_P \quad (2)$$

In this equation, the term *Energy* takes into account the number of non-local hydrophobic bonds, hydrophilic interactions, and interactions with the backbone. Also, the number of collisions (considered as penalties) and the penalty weight are considered in this term. This penalty is composed by the number of points in the 3D lattice that is occupied by more than one element (NC - number of collisions), multiplied by the penalty weight ($PenaltyValue$), as shown in Equation (3). $RadiusG_H$ and $RadiusG_P$ represent the gyration radius of the hydrophobic and hydrophilic side-chains, respectively. Radius of gyration is a measure of compactness of a set of points (in this case, the side-chains of the amino acids in the lattice). The more compact the set of points, the smaller the radius of gyration. Equation 4 shows how this measure is computed.

$$Energy = H - (NC \cdot PenaltyValue) \quad (3)$$

$$RG_{aa} = \sqrt{\frac{\sum_{i=1}^{N_{aa}}[(x_i - \overline{X})^2 + (y_i - \overline{Y})^2 + (z_i - \overline{Z})^2]}{N_{aa}}} \quad (4)$$

In this equation, x_i, y_i and z_i are the coordinates of the i-th side-chain of type "*aa*" of the protein, either hydrophobic (H) of polar (P); $\overline{X}$, $\overline{Y}$ and $\overline{Z}$ are the average of all x_i, y_i and z_i; and N_{aa} is the number of side-chains of type "*aa*".

In order to obtain a compact hydrophobic core, typical of globular proteins, the radius of gyration of the set of hydrophobic side-chains (RG_H) should be minimized, thus increasing the number of hydrophobic bonds between amino acids. Conversely, the maximization of the radius of gyration of the set of polar side-chains (RG_P) takes them to the outer side of the conformation. To obtain such effect, both terms are computed by Equations 5 and 6, where $maxRG_H$ is the value of the radius of gyration when the amino acids chain is completely stretched. Once computed $RadiusG_H$ and $RadiusG_P$, they are used in the fitness function shown in Equation 2.

$$RadiusG_H = maxRG_H - RG_H \quad (5)$$

$$RadiusG_P = \begin{cases} 1 & \text{if } (RG_P - RG_H \geq 0) \\ \frac{1}{1-(RG_P - RG_H)} & \text{else} \end{cases} \quad (6)$$

C. Genetic Operators and Improvement Strategy

Current literature presents many specialized genetic operators for the PFP and, in particular, some biologically-inspired operators. In this work we do not attempted to apply other genetic operators other than the regular two-point crossover and multibit mutation. Future work will focus on other operators.

When a GA gets trapped around a local maxima point in the search space, a decrement of population diversity usually takes place, mainly as a consequence of the intense local search provided by the crossover operator. This effect sometimes can be balanced by the action of the mutation operator. However, frequently, this is not enough and additional strategies are needed to avoid stagnation of the search.

When the population of the GA is concentrated around a local maxima, the only way to avoid useless computational effort is to escape from the current region, and redirects the GA to explore other regions of the search space. In this work we used the Decimation-and-Hot-Boot (DHB) strategy [11], [20], explained below.

During the evolution of the GA, the best individual of each generation is always maintained. An indirect evidence that the GA has stagnated is when the best individual does not improve for many generations. The strategy used verifies whether or not the best-so-far individual is improved from a given generation to the next one. If it is improved, a counter is zeroed, otherwise, it is incremented. When the counter reaches a predefined number of generations ($gen2decimate$), 50% of the population is decimated and substituted by individuals generated according to the same procedure done for the initial population (see section III-A). It is important to note that the best individual is always maintained during the DHB procedure.

The application of this strategy improves significantly the genetic diversity and allows the evolutionary process to continue for some more generations. Ultimately, the chances of finding even better solutions is improved. However, it should be taken into account that new individuals recently created by the DHB procedure probably will have low fitness values. If, in one hand, the genetic diversity is improved, on the other hand, the selective pressure is increased due to the large differences of fitness values between the individuals. It is well known that high selective pressure leads to premature convergence due to loss of genetic diversity. This is the opposite effect to what would be desired. Therefore, it is necessary to avoid high selective pressure during some generations just after the decimation. This is accomplished by decreasing the number of individuals that take part of the tournament selection (parameter $tourneysize$) to 2 during a fixed number of generations (parameter $gen2weakTourney$), and then returning to its original value.

IV. Computational Experiments and Results

All experiments reported in this work were run in a cluster of 31 computers with the same hardware and software configurations (Intel *core 2-quad* at 3 GHz, running Linux). The software was developed in ANSI-C programming language, using the Message Passing Interface (MPI) MPICH2 package for the communication between processes [2].

A. Benchmark sequences

In our experiments, 25 synthetic amino acids sequences were used as benchmark, as shown in Table II. These sequences had either 27, 31, 36 or 48 amino acids-long. Three groups of benchmarks were used: "Dill.*", first proposed by [24], "Unger273d.*" due to [22] and "S48.*" due to [25].

[2] Available at: http://www.mcs.anl.gov/research/projects/mpich2/

To the best of our knowledge, the "Unger273d.*" sequences were used for the first time by [3] for the 3DHP-SC model (results are shown in the column HnC), but the remaining were not yet tried for this model.

Only for comparison purposes, the maximum known number of hydrophobic contacts for "Dill.*", "Unger273d.*" and "S48.*" group of sequences using the 3DHP model are shown in the column (E) of the table, following results obtained by [24], [19] and [21], respectively. It is supposed that the optimal solution for these sequences, using the 3DHP-SC model, will have no less than the same number of contacts (of the 3DHP model), but, probably, more.

B. Control Parameters of the GA

There is no specific procedure for adjusting parameters of a GA. In this work we decided to do several preliminary experiments combining possible values for the parameters. For each combination, 100 independent runs were done using different random seeds, and the average results were compared.

The basic parameters of the GA were tested in the following ranges: tournament size of the selection procedure ($tourneysize$): 2%, 3%, 5%; probability of crossover operator (*pcross*): 70%, 80%, 90%; probability of the mutation operator (*pmut*): 2%, 5%, 8%. Also, the parameteres of the DHB strategy, $gen2decimate$ and $gen2weakTourney$, were tested in the ranges [300; 600] and [30; 60; 300; 600], respectively. The migration parameters, $Migrationgap$ and $MigrationRate$, were tested for the following values: [20; 30; 60; 90; 120; 150; 300; 600; 900] and [2; 3; 5], respectively. A total of 62 experiments were done.

By analyzing the average results we found the best set of parameters for the HPGA. They were used in all experiments reported in the next sections. The basic parameters of each island were: number of generations (3000), population size (500), two-points crossover probability (80%), multibit mutation probability (8%), selection method (stochastic tournament, $tourneysize$=3%). The specific parameters of the DHB strategy were: $gen2decimate$ = 300, $gen2weakTourney$ = 300. The migration policy parameters were set as: Migration gap = 120 generations, Migration Rate = 5 individuals, best and four random immigrants replace random individuals of the receiving population, topology with 4 islands connected by a unidirectional ring.

C. Optimization of the Energy Weights

In order to study the effect of weights of the energy function (Equation 1) in the quality of solution, a factorial experiment was done using one benchmark sequence of 27 amino acids (see section IV-A). The following values were tested: ϵ_{HH}: 10, 15; ϵ_{HP} and ϵ_{BH} : -5, -3; ϵ_{PP}, ϵ_{BP} and ϵ_{BB}: -1, 0, 1. Also, the joint effect of the penalty applied to the objective function as a consequence of collisions was also evaluated. Such penalty decreases the energy function by the product of the length of the sequence. This parameter ($PenaltyValue$) was tested for the values: 5, 7, and 10. All the possible combinations of the above mentioned values gives 36 different experiments. For each experiment, 30 independent runs were done with different initial random seeds. The average results were analyzed and the best performing set of parameters were used in the remaining experiments. Here, the quality was evaluated as the number of hydrophobic side-chains contacts. The optimized value for these parameters are: ϵ_{HH} = 10; ϵ_{HP} = ϵ_{BH} = -3; ϵ_{PP} = ϵ_{BP} = ϵ_{BB} = 1; $PenaltyValue$ = 10.

D. Decimation-and-Hot-Boot (DHB) Strategy

The HPGA was executed with and without the DHB strategy for 100 independent runs, keeping fixed all other parameters. Figure 2 shows a plot of the average fitness of the best individual ($Bestever$) for the two situations, at each generation. It is observed in this figure that the use of the DHB strategy leads to better individuals when compared with a HPGA without DHB, achieving, in this case, a gain that exceeds 10%. These results strongly suggests that the DHB strategy is advantageous for the GA and, therefore, it was used in all remaining experiments.

E. Benchmark Results and Discussion

Due to the stochastic nature of GA, the HPGA was run 100 times with a different initial random seeds for each of the 25 benchmark sequences, and results are shown in Table III. In this table, the first column identifies the sequence, the second, third and fourth columns identify, respectively, the generation in which the best individual was found, the average (± standard deviation) and maximum number of generations needed to find the best individual. Next, the average processing time for running in parallel (in seconds). The last two columns of the table show, respectively, the average value (± standard deviation), and the maximum number of non-local bonds between hydrophobic side-chains.

In this table it is observed that the HPGA needed, in average, less generations than the maximum allowed ($maxgen$ = 3000) to find the best-of-run solution. This fact suggests that a smaller number of generations could be used in future experiments with these benchmarks.

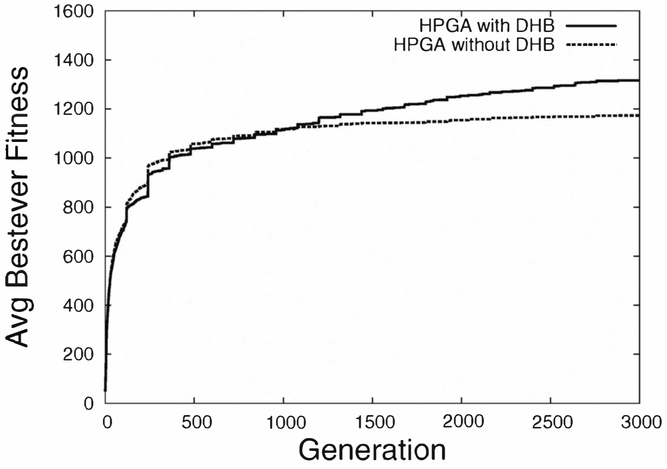


Figure 2. Performance of HPGA with and without the DHB strategy.

Table II

BENCHMARK SEQUENCES FOR 3DHP-SC MODEL: n INDICATES THE NUMBER OF AMINO ACIDS OF THE SEQUENCE, E THE MAXIMUM NUMBER OF NON-LOCAL HYDROPHOBIC CONTACTS FOR THE 3DHP MODEL, AND HnC IS THE EQUIVALENT FOR THE 3DHP-SC MODEL, HOWEVER CONSIDERING THE CONTACTS BETWEEN HYDROPHOBIC SIDE-CHAINS.

Reference	n	*HP Chain*	E	HnC [3]
Dill.1	27	$HP^4H^4P(PH)^3H(HP)^2PH^2P^2H$	16	–
Dill.2	27	$HP^3H^4(PH)^2HP^3HPH(HP)^2P^2HP$	15	–
Dill.3	27	$HPH^2(PPHH)^2H(HPPP)^2H^3P^2H$	16	–
Dill.4	31	$(HHP)^3H(HHHHHPP)^2H^7$	28	–
Dill.5	36	$PH(PPH)^{11}P$	14	–
Unger273d.1	27	$(PH)^3H^2P^2(HP)^2P^{10}H^2P$	9	10
Unger273d.2	27	$PH^2P^{10}H^2P^2H^2P^2HP^2HPH$	10	12
Unger273d.3	27	$H^4P^5HP^5H^3P^8H$	8	11
Unger273d.4	27	$H^3P^2H^4P^3(HP)^2PH^2P^2HP^3H^2$	15	18
Unger273d.5	27	$H^4P^4HPH^2P^3H^2P^{10}$	8	11
Unger273d.6	27	$HP^6HPH^3P^2H^2P^3HP^4HPH$	11	13
Unger273d.7	27	$HP^2HPH^2P^3HP^5HPH^2(PH)^3H$	13	16
Unger273d.8	27	$HP^{11}(HP)^2P^7HPH^2$	4	6
Unger273d.9	27	$P^7H^3P^3HPH^2P^3HP^2HP^3$	7	9
Unger273d.10	27	$P^5H(HP)^5(PHH)^2PHP^3$	11	14
S48.1	48	$HPH^2P^2H^4PH^3P^2H^2P^2HPH^3(PH)^2HP^2H^2P^3HP^8H^2$	32	–
S48.2	48	$H^4(PHH)^2H^3(PPH)^2HP^2HP^6(HPP)^2PHP^2H^2P^2H^3PH$	34	–
S48.3	48	$(PH)^2HPH^6P^2(HP)^2(PH)^2(HP)^3(PPH)^2HP^2H^2P^2(HP)2PHP$	34	–
S48.4	48	$(PH)^2HP^2HPH^3P^2H^2PH^2P^3H^5P^2HPH^2(PH)^2P^4HP^2(HP)^2$	33	–
S48.5	48	$P^2HP^3HPH^4P^2H^4(PHH)^2HP(PH)^3P^2HP^5(PHH)^2PH$	32	–
S48.6	48	$H^3P^3H(HP)^2(HHP)^3HP^7(HP)^2PHP^3HP^2H^6PH$	32	–
S48.7	48	$PHP^4HPH^3(PH)^2H^3(PHH)^2P^3(HP)^2P^2H^3(PPHH)^2P^3H$	32	–
S48.8	48	$(PHH)^2HPH^4P^2H^3P^6HPH^2P^2H(HP)^2P^2H^2(PH)^3HP^3$	31	–
S48.9	48	$(PH)^2P^4(HP)^3(PH)^2H^5P^2H^3PHP(PH)^2HP(PH)^2H^2P^4H$	34	–
S48.10	48	$PH^2P^6H^2P^3H^3PHP(PH)^2(HPP)^3H^2P^2H^7P^2H^2$	33	–

If the GA was able to achieve the maximum number of non-local bonds (last column of table III) in all runs, its efficiency would be 100%. Considering all the benchmark sequences, the proposed HPGA achieved an average efficiency that exceeds 80%. This value can considered very good, taking into account the differences between instances, the stochastic nature of a GA and the number of parameters to be adjusted. Therefore, it can be inferred that the proposed GA performs consistently.

It is also observable that the total processing time is dependent on the length of the sequence, possibly growing exponentially with the number of amino acids. This fact suggests that the HPGA will lose performance for larger sequences.

Finally, the present version of the GA is significantly more efficient than a previous work [3]: for the benchmarks "Unger273d.*" it obtained better results in 7 out of 10 cases (comparing last column of Tables II and III).

For some of the benchmark sequences the best conformation found is shown in Figures 3(a), 3(b), 3(c), 3(d), 3(e), 3(f) and 3(g). It is possible to observe in these foldings that a compact hydrophobic core is formed, partially surrounded by amino acids with polar side-chains. This type of conformation, typical of globular proteins, was expected as consequence of the fitness function. This fact suggests that the proposed fitness function is adequate for the PFP problem using the 3D-HP side-chain model, mainly because the final results are capable to mimic some biological properties of real proteins during folding.

V. CONCLUSIONS

This work proposed a hierarchical parallel genetic algorithm for the protein folding problem using the 3DHP-side-chain model. To date, there is only a previous version [3] to compare with and these are the best results found for the 3DHP-SC model. Therefore, an important contribution of this work are the results regarding this issue. However, closely observing figure 3(g) it is possible to notice that small (but relevant) improvements could be further done with that folding. Such improvements could be achieved by means of some local search strategy, to be investigated in a future work.

In this work we also studied the effect of the energy weights in the performance of the algorithm. Therefore, another relevant contribution is the set of optimized weights for the 3DHP-SC model, not yet available in the literature.

Future work will also investigate parallel versions of other evolutionary computation approaches, such as Ant Colony Optimization (ACO) [10], Particle Swarm Optimization (PSO) [13], Artificial Bee Colony (ABC) [12] and Firefly Algorithm (FA) [23], so as to compare with the HPGA presented in this study.

Overall results are good and very promising to suggest the continuity of the work, representing a significant increment over [3], in quality and extension. This work also offered new reference values for three sets of benchmark sequences that

Table III

RESULTS OF THREE GROUPS OF BENCHMARK SEQUENCES FOR THE 3DHP-SC MODEL.

Reference	Generation		avg T_p(s)	HnC	
	best	avg ± stdev		avg ± stdev	max
Dill.1	2160	2151.80 ± 687.28	573.50	17.42 ± 1.86	21
Dill.2	2904	2084.25 ± 770.10	566.59	14.78 ± 1.56	19
Dill.3	2880	2378.29 ± 583.31	566.68	18.00 ± 1.79	23
Dill.4	2760	2154.31 ± 603.6	749.10	32.81 ± 2.40	41
Dill.5	2880	2104.10 ± 741.37	1141.11	11.29 ± 1.27	14
Unger273d.1	2400	2153.13 ± 702.58	572.80	10.06 ± 1.18	**12**
Unger273d.2	2640	2025.67 ± 725.73	569.10	11.89 ± 0.93	**13**
Unger273d.3	2880	1996.71 ± 740.37	569.48	10.71 ± 0.94	**13**
Unger273d.4	1778	2405.63 ± 498.65	566.93	17.79 ± 2.20	**22**
Unger273d.5	2779	2340.37 ± 438.18	560.04	10.83 ± 1.04	**13**
Unger273d.6	1680	2126.62 ± 612.23	568.20	11.28 ± 1.29	**14**
Unger273d.7	1680	2019.89 ± 763.72	566.53	12.57 ±1.50	16
Unger273d.8	1560	1523.00 ± 817.37	569.05	4.68 ± 0.85	6
Unger273d.9	2400	1818.24 ± 846.19	568.18	8.06 ± 1.03	**10**
Unger273d.10	2400	2068.67 ± 871.71	568.99	11.08 ± 1.23	14
S48.1	2882	2467.10 ± 593.50	2640.50	26.24 ± 3.13	32
S48.2	2520	2523.32 ± 373.44	2640.97	26.68 ± 3.07	32
S48.3	2880	2367.69 ± 619.59	2644.03	24.63 ± 3.32	30
S48.4	2760	2541.63 ± 469.90	2647.95	25.75 ± 3,47	35
S48.5	2964	2340.37 ± 723.44	2647.43	26.37 ± 2.87	31
S48.6	2780	2275.26 ± 658.95	2644.14	26.29 ± 3.04	33
S48.7	2144	2319.56 ± 687.14	2645.08	24.63 ± 3.56	32
S48.8	2743	2335.81 ± 596.25	2647.55	25.94 ± 2.89	31
S48.9	2084	2378.71 ± 516.47	2650.81	26.12 ± 2.99	32
S48.10	1800	2135.00 ± 480.95	2655.54	27.75 ± 2.89	33

can be used in the future by other researchers and optimization methods.

We believe that this work provides a contribution to this area of research because of three factors: deeper exploring the 3DHP side-chain model suggesting optimized weights for the energy function, providing benchmark results useful for comparison with other approaches, and modeling an efficient HPGA for the PFP. Further work will focus on generating more benchmark results, as well as the development of biologically inspired genetic operators and local-search strategies.

ACKNOWLEDGEMENTS

Authors would like to thanks the Brazilian National Research Council – CNPq, for the financial support under grants 550977/2007-4, 481207/2007-0 and 309262/2007-0 to H.S.Lopes and a graduated scholarship to C.M. Vargas Benítez.

REFERENCES

[1] C.B. Anfinsen. Principles that govern the folding of protein chains. *Science*, (181), 1973.

[2] J. Atkins and W.E. Hart. On the intractability of protein folding with a finite alphabet. *Algorithmica*, 25(2-3):279–294, 1999.

[3] C.M.V. Benítez and H.S. Lopes. A parallel genetic algorithm for protein folding prediction using the 3DHP side-chain model. In *Proc. IEEE Congr. on Evolutionary Computation*, pages 1297–1304, Piscataway, USA, 2009.

[4] C.M.V. Benítez and H.S. Lopes. Protein structure prediction with the 3D-HP side-chain model using a master-slave parallel genetic algorithm. *Journal of the Brazilian Computer Society*, 16:69–78, 2010.

[5] H.M. Berman, J. Westbrook, Z. Feng, and G. Gilliland et al. UniProt archive. *Nucleic Acids Research*, 28(1):235–242, 2000.

[6] R. Bianchini and C. Brown. Parallel genetic algorithms on distributed-memory architectures. In *Proc. 6^{th} Conf. North American Transputer Users Group on Transputer Research and Applications (NATUG-6)*, pages 67–82, 1993.

[7] E. Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Springer, New York, USA, 2000.

[8] The UniProt Consortium. The universal protein resource (UniProt) 2009. *Nucleic Acids Research*, 37:D169–D174, 2009.

[9] K.A. Dill, S. Bromberg, K. Yue, and K.M. Fiebig et al. Principles of protein folding - a perspective from simple exact models. *Protein Science*, 4(4):561–602, 1995.

[10] M. Dorigo and T. Stützle. *Ant Colony Optimization.* MIT Press, Cambridge, MA, USA, 2004.

[11] F. Hembecker, H.S. Lopes, and W. Godoy Jr. Particle swarm optimization for the multidimensional knapsack problem. *Lecture Notes in Computer Science*, 4331:358–365, 2007.

[12] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, Department Computer Engineering Department, Erciyes University, 2005.

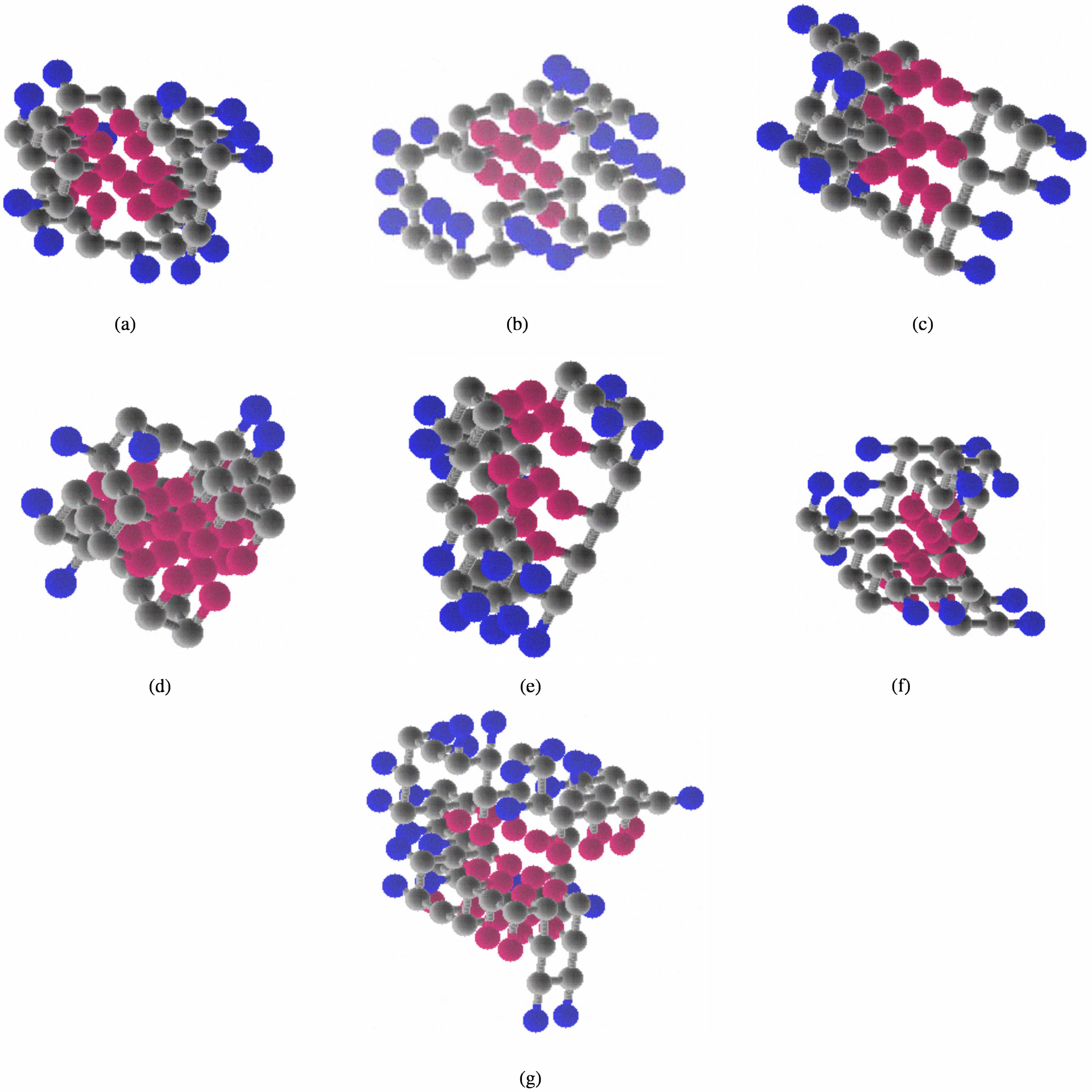
[13] J. Kennedy and R. Eberhart. Particle Swarm Optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, Piscataway, NJ, USA, 1995. IEEE Computer Society.

[14] N. Krasnogor, W.E. Hart, J. Smith, and D.A. Pelta. Protein structure prediction with evolutionary algorithms. In *Proc. Genetic and Evolutionary Computation Conf.*, pages 1596–1601, 1999.

[15] M.S. Li, D.K. Klimov, and D. Thirumalai. Folding in lattice models with side chains. *Computer Physics Communications*, 147(1):625–628, 2002.

[16] H.S. Lopes. Evolutionary algorithms for the protein folding problem: A review and current trends. In *Computational Intelligence in Biomedicine and Bioinformatics*, volume I, pages 297–315. Springer-Verlag, Heidelberg, 2008.

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



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 3. Best 3D folding for sequences found for Dill.2 (a), Unger273d.3 (b), Unger273d.4 (c), Dill.4 (d), Unger273d.7 (e), Dill.3 (f) and S48.10 (g). Blue balls represent the polar residues and Red ones represent the hydrophobic residues. The backbone and the connections between elements are shown in gray.

[18] D.L. Nelson and M.M. Cox. *Lehninger Principles of Biochemistry.* W.H. Freeman, 5^{th} edition, 2008.

[19] A.L. Patton, W.F. Punch III, and E.D. Goodman. A standard GA approach to native protein conformation prediction. *Proc. 6^{th} Int. Conf. on Genetic Algorithms*, pages 574–581, 1995.

[20] M.P. Scapin and H.S. Lopes. A hybrid genetic algorithm for the protein folding problem using the 2D-HP lattice model. In A. Yang, Y. Shan, and L.T. Bui, editors, *Success in Evolutionary Computation*, number 205-224, pages 205–224. Springer, Heidelberg, 2007.

[21] C. Thachuk, A. Shmygelska, and H.H. Hoos. A replica exchange Monte Carlo algorithm for protein folding in the HP model. *BMC Bioinformatics*, 8, 2007.

[22] R. Unger and J. Moult. A genetic algorithm for 3D protein folding simulations. *Proc. 5^{th} Ann. Int. Conf. on Genetic Algorithms*, pages 581–588, 1993.

[23] Xin-She Yang. *Nature-Inspired Metaheuristic Algorithms.* Luniver Press, York, UK, 2008.

[24] K. Yue and K.A. Dill. Sequence-structure relationships in proteins and copolymers. *Physical Review E*, 48(3):2267–2278, 1993.

[25] K. Yue, K. Fiebig, and P. Thomas et al. A test of lattice protein folding algorithms. *Proceedings of the National Academy of Sciences of USA*, V. 91:581–588, 1994.