12

Evolutionary Algorithms for the Protein Folding Problem: A Review and Current Trends

Heitor Silvério Lopes

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

12.1 Introduction

Proteins are complex macromolecules that perform vital functions in all living beings. They are composed of a chain of amino acids. The biological function of a protein is determined by the way it is folded into a specific tri-dimensional structure, known as native conformation. Understanding how proteins fold is of great importance to Biology, Biochemistry and Medicine. Considering the full analytic atomic model of a protein, it is still not possible to determine the exact tri-dimensional structure of real-world proteins, even with the most powerful computational resources. To reduce the computational complexity of the analytic model, many simplified models have been proposed. Even the simplest one, the bi-dimensional Hydrophobic-Polar (2D-HP) model (see Sect. 12.2.2), was proved to be intractable due to its NP-completeness. The current approach for studying the structure of proteins is the use of heuristic methods that, however, do not guarantee the optimal solution. Evolutionary computation techniques have been proved to be efficient for many engineering and computer science problems. This is also the case of unveiling the structure of proteins using simple lattice models.

In this work the nature of the models used for the protein folding problem is reviewed, with special emphasis on discrete models. Also, we analyze how evolutionary computation techniques have been applied to solve it. Amongst these techniques, there are many different variants of genetic algorithms, besides ant colony optimization, differential evolution and artificial immune systems.

This chapter is structured as follows: the remaining of this section introduces some basic aspects of amino acids and proteins, and presents the protein folding problem. Sect. 12.2 presents the several models for protein folding with special emphasis on a specific discrete model: the hydrophobic-polar. Sect. 12.3 is dedicated to the several computational approaches for the protein folding problem, from molecular dynamics and approximation algorithms to several evolutionary computation algorithms. Next, Sect. 12.4 presents challenging issues that limit current research. Finally, in Sect. 12.5 current trends for future research and the conclusion are presented.

T.G. Smolinski et al. (Eds.): Comp. Intel. in Biomed. & Bioinform., SCI 151, pp. 297–315, 2008.
springerlink.com

12.1.1 Amino Acids and Proteins

The basic structure of an amino acid consists of a carbon atom (C_α) connected with an amino group (NH_2), a carboxyl group (COOH) and a side-chain. The only difference between amino acids is due to the composition of their side-chain. There are 20 standard amino acids. According to the physical properties of the side-chain, amino acids can be classified according its polarity and acidity/basicity. Such classification leads to a hydrophilic (polar) or hydrophobic (nonpolar) character of the amino acid. The distribution of hydrophilic and hydrophobic amino acids along the protein ultimately determines structure of the protein.

The sequence of amino group, C_α and carboxyl group of an amino acid bounded with the following is known as backbone of a protein. There are three main levels of organization of the structure of a protein: primary, secondary and tertiary structures. The primary structure of a protein or polypeptide chain is its linear sequence of amino acids, represented by a string of letters. Some specific regions of the primary structure can fold into known tri-dimensional structures, such as α-helices or β-sheets. These structures are known as secondary structures. The spatial representation of the protein is called tertiary structure. The shape into which a protein naturally folds is known as its native state, or native conformation. For some particular proteins, tertiary structures can be combined to form a super-structure known as quaternary structure.

The tertiary structure of a protein, or the quaternary structure of its complexes, is of particular interest, since it defines the biological function of the protein. The most effective method for unveiling the structure of real proteins is using nuclear magnetic resonance spectroscopy or X-ray crystallography. It is estimated that the human body has around 100,000 different proteins, but a only a small portion of them have its structure known. The Protein Data Bank (PDB) [7] ($http://www.pdb.org$) is the repository for structural data of proteins. Currently, it holds structural information of almost 50,000 proteins. However, the amount of known proteins which structure is unknown is much larger, thus justifying the use of computational methods for this purpose. Therefore, this is an important research area in Bioinformatics and Computational Biology.

12.1.2 Protein Folding

The Protein Structure Prediction (PSP) problem can be defined as determining the final tri-dimensional structure of a protein by using only the information about its primary structure. On the other hand, the Protein Folding Problem (PFP) is understood as being the discovery of the pathways by which a protein is folded into its natural conformation, during its synthesis [34]. However, in the current literature those two terms are frequently used with no distinction, usually meaning only the first issue. A computational approach to predict the structure of a protein demands a model that represents it abstractly, in a given level of details. Basing on well-established thermodynamical laws, the prediction of the structure of a protein is modelled as the minimization of the corresponding free-energy with respect to the possible conformations that a protein is able to attain.

The minimization of this free-energy is the most important factor that drives the construction of the structure of a protein. Formally, the native conformation of a protein is defined as that in which the free-energy is minimal. According to [61], a computational model that obeys this principle must have the following features:

- A model of the protein, defined by a set of entities representing atoms and connections among them;
- A set of rules defining the possible conformations of the protein;
- A computationally feasible function for evaluating the free-energy of each possible conformation.

The amount of details of the structure modelled depends on the choices done about the model (see Sect. 12.2). For instance, a protein could a have a spatial representation of all its atoms, all all its atoms but hydrogen, only the backbone without the side-chain, or as simple hydrophobic-polar elements embedded in a lattice.

12.2 Free-Energy Models

12.2.1 Analytical Models

An analytical model has a detailed description of the tri-dimensional structure of a protein, including information about all its individual atoms [61]. A protein can be viewed as a collection of atoms connected each other. Therefore, to specify the tertiary structure of a protein, it is possible to establish values for angles, lengths and torsions of the connections among atoms in the structure. To reduce the inherent complexity of this model, some atoms could be disregarded or even grouped into larger elements, and treated as equivalent single atoms by the model. Obviously, such reduction decreases the visual equivalence between the model and a real protein, for a given conformation.

The free-energy function in an analytical model is frequently specified by parameters representing the individual contributions of atoms. For atoms connected each other, such parameters depend on the length, angles and torsions of the connection. For those atoms that are not directly connected each other, the parameters depend on physical forces (i.e., Coulomb and van der Walls forces) or statistical information inferred from known structures. However, using a detailed description of the structure of a protein and many parameters in the free-energy function, it is computationally hard to find an optimized solution for the prediction of the structure of a protein. For instance, analytical models were presented by [10, 25, 56, 57].

12.2.2 Discrete Models

The difficulty in using analytical models motivated researchers to develop simpler discrete models that allow a large number of computational simulations necessary

to find optimal or quasi-optimal solutions for the PFP [10, 23]. The easiest way to limit the complexity of an analytical model is to limit the range of lengths, angles and torsions allowed in the model, and use predefined sets of values. Usually, these allowed values are obtained from known real-world structures [60, 61]. The simplest class of models for the PFP is known as lattice models. In these models, a protein is modelled as a sequence of simple elements, representing the amino acids, embedded in a lattice. The connection angles between amino acids are restricted by the lattice structure in the plane (2D) or in the space (3D). In a valid conformation, a given position in the lattice can be occupied by, at most, one amino acid, and adjacent amino acids in the sequence must occupy adjacent positions in the lattice. The free-energy of a conformation is defined as a function of the number of adjacent amino acids in the structure which are non-adjacent in the sequence. This is known as non-local bonds [22, 61] or H-H contacts. Although square and cubic lattices are the most popular, there are implementations that use other type of lattices, such as triangular [46, 63] and hexagonal [37].

Despite the simplicity of lattice models, both 2D and 3D HP models have some behavioral equivalency with real-world proteins [22, 23, 24, 61]. Also, the computational treatment of such models are much more convenient, when compared with analytical models and, for some instances, the exhaustive enumeration of the possible conformations can be done. These properties have made lattice models very popular. However, the main drawback of lattice models (and, in special, of HP models - see Sect. 12.2.2) is the difficulty in representing clear secondary structures in the folding [33].

The Hydrophobic-Polar Model

This model was introduced by [42] and it is the most known and studied discrete model for the PFP. The Hydrophobic-Polar (HP) model is the simplest possible model and, in most cases, uses a square (2D) or cubic (3D) lattice. Notwithstanding, even being simple, the PSP was proved to be NP-hard using this model, that is, there is no polynomial-time algorithm to solve it, either the 2D version [16, 55, 57, 73] or the 3D one [3, 6]. This fact has motivated the development of many heuristic approaches, such as in [8, 10, 22, 50, 52, 61, 70].

The HP model is based on the assumption that the major contribution to the free-energy of the native conformation of a protein is due to interactions between hydrophobic amino acids, which tend to be grouped in the inner part of the spatial structure, while the hydrophilic (polar) amino acids tend to stand more outside, thus protecting the hydrophobic amino acids from contact with the environmental solvent. For simplicity, the 20 standard amino acids are divided in either hydrophobic (H) or polar (P), based on experimental results [45]. Therefore, the primary structure of a protein is a string defined over the binary alphabet $\{H, P\}$. Although several different hydrophobicity scales can be found in the current literature, there is still no consensus about a standard translation table between the 20-letters amino acids into a simple $\{H, P\}$ alphabet. To circumvent this problem, some studies suggest the use of extended alphabets,

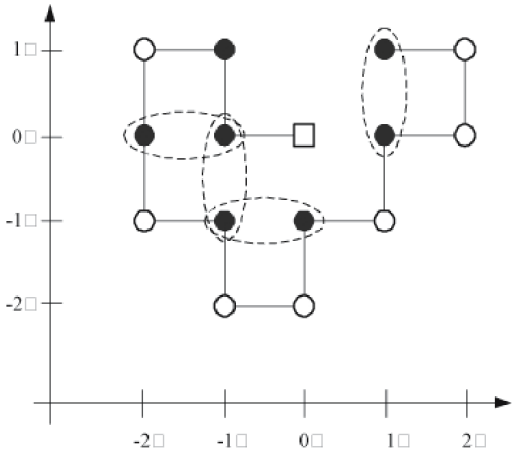

Fig. 12.1. A hypothetical conformation of 15 amino acids using the 2D-HP model

in which amino acids are converted to symbols more properly related to their physical and chemical properties [5, 48].

The $\{H, P\}$ string representing a protein is then embedded in a 2D or 3D lattice. Adjacent amino acids of the sequence are also adjacent in the lattice and, for a valid conformation, no point of the lattice can be occupied by more than one amino acid. The free-energy of a conformation is inversely proportional to the number of non-local bonds, as defined before. It is worth to mention that a non-local bond only takes place when a pair non-adjacent amino acids of the sequence lie in adjacent positions in the lattice. Consequently, minimizing the free-energy is equivalent to maximize the number of hydrophobic non-local bonds.

Figure 12.1 presents a conformation of polypeptide with 15 amino acids using the 2D-HP model. Black and white dots represent, respectively, the hydrophobic and hydrophilic amino acids. The square dot is the first amino acid of the sequence. The chain is connected by solid lines, and the bonds are represented by dotted lines. For this conformation there are 4 H-H contacts.

A simple free-energy function of a conformation, suggested by [44], is represented in Eq. (12.1):

$$E = \sum_{i<j} e_{r_i r_j} \Delta(r_i - r_j) \tag{12.1}$$

where: $\Delta(r_i - r_k) = 1$, if amino acids r_i and r_j have a non-local bond, or $\Delta(r_i - r_k) = 0$, otherwise. Depending on the type of contact between amino acids, the energy will be e_{HH}, e_{HP} or e_{PP}, corresponding to H-H, H-P or P-P contacts, respectively. According to [44], these energy parameters satisfy the following physical constraints:

1. Compact conformations have energy levels smaller than any other non-compact conformations;

2. Hydrophobic amino acids tend to be buried as inside as possible in the conformation. This is expressed by the relation $e_{PP} > e_{HP} > e_{HH}$ that decreases the energy of conformations in which the hydrophobic amino acids are hidden from the water solvent;
3. Amino acids of different types tend to get apart. This is expressed by the relation $2e_{HP} > e_{PP} + e_{HH}$.

In the standard HP model, values for those parameters are: $e_{HH} = -1.0$, $e_{HP} = 0$ and $e_{PP} = 0$ [42]. However, [44] suggested $e_{HH} = -2.3$, $e_{HP} = -1$ and $e_{PP} = 0$, since they satisfy the above conditions. According to them, results are not too sensitive to values of e_{HH}, provided those conditions are satisfied.

Other Discrete Models

Besides the popular HP model, there are other discrete models for the PFP in which particular biological properties were explored:

- Lattice Polymer Embedding (LPE): this model was proposed by [73] and is based in a cubic lattice, similarly to the HP model. Each pair of amino acids have an affinity coefficient and the energy function to be minimized is the sum, over all possible pairs of amino acids, of the product of the affinity coefficient by the distance between amino acids.
- Charged Graph Embedding (CGE): This model was proposed by [27] and later used by [10]. It uses a 3D lattice and incorporates charges to the amino acids. Conformations allowed are not very realist because it considers bonds between every pair of amino acids in the chain and bonds are allowed to cross each other. On the other hand, the influence of a given amino acid on another one disappears when the Euclidean distance between them exceeds a critical value.
- Perturbed Homopolymer (PH): this model was suggested by [64] and reviewed by [67], and later used by [22]. This model does not take into account only the interactions between hydrophobic amino acids, but favors connections between amino acids of the same type (that is, H-H and P-P).
- Helical-HP model: it was presented by [72] and reviewed by [22]. This model considers only a 2D lattice and includes two types of interactions: non local interactions between hydrophobic amino acids, and local interactions represented by a propensity to form helices (helical propensity). [11] has extended this model taking into consideration the effect of hydrogen bonds in regular secondary structures (in both α-helices and β-sheets).
- Tangent Spheres Side Chain model (HP-TSSC): introduced by [31], it uses the basic HP model, but does not embed amino acids in a lattice. In this model, a protein is represented by a graph that is transformed in a set of tangent spheres with equal radius, for both the backbone and the side chains. This model is an important contribution to off-lattice models for protein folding.
- HPNX model: this is a variation of the HP model in which the alphabet is extended to more letters [5, 9]. For instance, polar amino acids can be divided into three categories: positive charge (P), negative charge (N) and

neutral (X). Therefore, the standard 20 amino acids is translated into the $\{H, P, N, X\}$ alphabet according to their physical and chemical properties and the energy of a conformation is computed using a matrix of energy potentials between every pair of contacts. Usually, $e_{HH} = -4.0$, $e_{PP} = e_{NN} = 1.0$, $e_{NP} = -1.0$ and for the remaining type of contacts the energy is null.

12.3 Computational Approaches for the PFP

12.3.1 Molecular Dynamics

This method, also known as *ab initio* [29, 58], seems to be the most realistic approach to simulate the folding of real proteins. The term *ab initio* means to start "from the beginning", without using previous knowledge about the structure of the protein. To do so, the basic idea is to simulate the movements of each atom of a protein, as well as of the water that surrounds it, as a function of time. The initial thermal energy of the system is established and atoms are enabled to move according to the rules of classical mechanics. The energy of a conformation is adapted to take into account all forces, accelerations and velocities to which each atom is submitted along time. Aiming at making the movement of the atoms the more realistic as possible, a very small time step is defined, such as 10^{-15} sec. For each time step the energy function is recomputed [75]. Even using supercomputers, the number of mathematical operations necessary to simulate the folding is so high that makes this methodology unfeasible even for very of small proteins. This type of simulation can be useful only for studying the behavior of the folding during very short periods of time, several orders of magnitude smaller than the time necessary to fold a real-world protein. However, it is believed that this method is potentially powerful to produce results according to the dynamic properties observed during the folding of real-world proteins [4], although it cannot be assured that it will converge to the native conformation. A full review of this methodology can be found in [43].

12.3.2 Approximation Algorithms

An approximation algorithm is a computational procedure capable of finding quasi-optimal solutions for specific problems (or specific instances), with a given predefined warranty of performance (regarding the optimal solution). According to [57], such class of algorithms can be useful for the PFP since they can find valid conformations somewhat near to the native conformation of a protein (provided the guaranteed maximum error is small). In a further step, the free-energy value of this approximated solution can be used as an upper-bound for another algorithm focused on local search. The main drawback of approximation algorithms is the need for a formal proof of its lower-bound performance.

Possibly, the first approximation algorithms devised for the PFP were proposed by [30, 31] using 2D- and 3D-HP models. Later, many other algorithms were proposed for different energy models and geometry of lattices [31, 32, 53, 54].

12.3.3 Genetic Algorithms

The minimization of the free-energy in discrete models frequently leads to a hard optimization problem. Despite the simplicity the usual free-energy function, based on the maximization of non-local H-H bonds, it leads to a multimodal search space. This search space is characterized by a large number of invalid solutions (corresponding to conformations in which more than an amino acid occupy a position in the lattice) and many local minima (corresponding to different conformations with the same number of non-local H-H bonds). These characteristics of the search space make it a hard problem for conventional optimization methods.

Amongst the many computational approaches for the PFP, certainly the most used is the genetic algorithm (GA), possibly due to its simplicity and efficiency in finding good solutions in large and complex search spaces. This of a GA in combining local features into a global solution makes it particularly appealing for the PFP.

A protein can have well-defined secondary structures, such as α-helices or β-sheets. In most cases, important secondary structures can be identified in the primary structure as motifs. Some motifs have structures independent of its interaction with the remaining molecule, and they can be viewed as building blocks. In a similar way, the crossover operator of a GA works by recombining hopefully useful blocks to form solutions of increasing quality, thus providing a way to recycle partial solutions.

There are two basic issues for applying a GA for a given optimization problem: how the variables of the problem are encoded, and how the quality of a solution is measured. The first issue is the representation problem, and the latter, the evaluation problem. All other issues raised in an implementation, although important, are secondary.

Encoding

When using a GA for the PFP, the way conformations of the protein are represented has a great importance on the dynamics and efficiency of the algorithm. Basically, one can devise three ways of representing a folding [40, 62]:

- Distance matrix: this encoding system describes a structure by means of a square matrix in which cells represent the distance between amino acids. This encoding system is rarely used in the literature [62].
- Cartesian coordinates: in this approach, a folding is described by a vector of elements representing the position of the amino acids of a sequence in the plane $\{x_i, y_i\}$ or in the space $\{x_i, y_i, z_i\}$. In general, this approach is not the most adequate for population-based algorithms (such as the evolutionary computation ones), since identical (or similar) structures can have completely representing vectors.
- Internal coordinates: a given conformation is represented as a set of movements of an amino acid relative to its predecessor in the chain. This is the

most usual representation approach found in evolutionary algorithms for the PFP, and two types of internal coordinates can be used:

- Absolute internal coordinates: they are based on the orientation of the axes the lattice in which the folding is embedded (either 2D or 3D). This encoding system is defined by the following set: $\{N, S, E, W, F, B\}$, corresponding to movements north, south, east and west (in the plane), and forward and backward (in the space).
- Relative internal coordinates: this encoding system defines the position of the next amino acid of the chain relative to the position of the preceding one in the lattice. The possible set of movements are: $\{F, L, R, U, D\}$, corresponding to forward, left, right, up and down, always having the previous position as reference. This encoding system has an important drawback: the initial population of a GA is randomly generated and, as a consequence, individuals will have an increased number of collisions in the structure (invalid conformations). This is specially true for proteins with an increased number of amino acids.

For instance, the conformation shown in figure 12.1 corresponds to the sequence $\{PHHPHPHPPHPHPPH\}$ and can be represented using:

- Relative internal coordinates: $\{LRLLFLRLLRLRLL\}$,
- Absolute internal coordinates: $\{WNWSSESENENENW\}$,
- Cartesian coordinates: {(0,0);(-1,0);(-1,1);(-2,1);(-2,0);(-2,-1); (-1,-1); (-1,-2);(0,-2);(0,-1);(1,-1);(1,0);(2,0);(2,1);(1,1)}.

A study of the two internal relative coordinates was done by [40], using different types of lattices. They concluded that, for square and cubic lattices, relative internal coordinates may lead a genetic algorithms to results much better than those that could be obtained using absolute internal coordinates. However, there are some authors who obtained satisfactory results using absolute coordinates for small chains [17]. For a triangular lattice, both types of coordinates have the same performance.

There are two restrictions to be satisfied for a valid conformation: there should be no collisions (a given point in the lattice should be occupied by at most one amino acid), and all adjacent amino acids of the sequence must be adjacent in the lattice. This last restriction is implicit in the encoding when using internal coordinates, but not when using Cartesian coordinates. To deal with the first restriction using internal coordinates there are two basic approaches:

- Delete invalid conformations that appear during the evolutionary cycle. This is the simplest way to deal with this issue, but, possibly, not the best one. When a protein is folded in a valid (but not optimal) conformation, the pathway to another valid conformation of smaller energy may be not achievable unless some invalid conformations are permitted in intermediary steps.
- Allow invalid conformations in the population and apply penalties. This approach is usual when using evolutionary algorithms for constrained problems. The genetic material present in some unfeasible solutions can be recombined further in the evolutionary cycle so as to form feasible and, hopefully, better

solutions. For the PFP there are two ways for applying penalties to invalid conformations: considering the number of pairs of amino acids that stand on the same point in the lattice, or considering the number of lattice points that have more than one amino acid in it. To date, it is not clear which of the two methods will give better results. Another somewhat different approach is due to [59] who suggest that hydrophobic amino acids that are in lattice points already occupied by other amino acids should not contribute to the free-energy function. that have more than two amino acids. This is an indirect way to apply a penalty to invalid conformations.

For off-lattice models, the encoding is somewhat straightforward. For instance, [20] represented a protein by means of internal angular coordinates of the atoms of the main chain. The torsion angles of the C_α (namely, ϕ and ψ) were restricted to a small set of possible values, and were sufficient to represent the topology of the main chain for a large number of proteins with known structure. Therefore, using this kind of model, a chromosome can be encoded with integer [14] or binary [20, 21, 60] values representing those angles. On the other hand, [66] used a chromosome of real-valued genes for representing the same angles.

Fitness Function

There are many variations on the fitness function, and they are based on the model used (see Sect. 12.2.2).

For instance, [11] has proposed the use of an extra term to the Eq. (12.1), named secondary-structure-favored energy term, that considers the energy between hydrogen bonds formed by secondary structures. Also, [50] proposed a fitness function having three terms: the first is the regular free-energy function of the HP model and the other two are based on the concept of radius of gyration. The radius of gyration is computed separately for hydrophobic and for polar amino acids. Maximizing the radius of gyration of hydrophobic amino acids means that they are pushed towards the inner part of the conformation, while maximizing the radius of gyration of polar contacts means pushing them towards outside. This concept was used to force more compact and globular-like conformations.

Other variations can be found: [17] used a weighted sum of the number of H-H contacts, the number of H-P contacts and the number of hydrophobic-solvent contacts. They argue that this fitness function is more natural from the biological point of view, since it may be preferable for a hydrophobic amino acid to have a contact with a polar amino acid than to be in direct contact with the solvent.

Most approaches in the literature use some fitness function based on the number of H-H contacts, inherent to the HP model. However, the main criticism of this simple approach is that hydrophobic iteration alone is not sufficient to induce regular structures during folding, as pointed by [11].

Genetic Operators

For most implementations, the regular crossover and mutation operators have been used as part of a larger set of specialized operators.

Regarding the regular crossover, there are implementations using 1-point, 2-points and uniform variants. Although there is no consensus about which crossover type gives the best results, the traditional 1-point crossover is less disruptive and tends to keep larger schemata. Therefore, the more folded a conformation, the more the 1-point crossover seems to be appropriate. A different approach, known as systematic crossover, was proposed by [38]. In this case the best individual is always one of the parents selected for crossover and all possible crossover points are tried, generating a number of individuals. The two best offsprings are maintained in the population.

Some special types of mutation were also proposed. For instance, [15, 68, 74] proposed in-plane rotation, snake, out-of-plane rotation, crank shaft, kink and cornerchange, and [35] implemented diagonal move and tilt move. All these mutation operators aimed at producing different conformations by means of specific re-arrangement of the folding in the lattice.

Other researchers presented biologically-inspired operators such as the U-turn and Make-loops by [51]. These operators were meant to simulate the construction of stable secondary structures found in real folded proteins, such as α-helices and β-sheets.

Two special genetic operators were proposed by [15]: duplicate predator and brood selection. The first is aimed at maintaining diversity in the populations throughout generations by means of deleting duplicate individuals and is similar to the pioneer search strategy introduced by [38]. The latter generates a brood of offsprings from two parents, and the best descendent is kept. This procedure is a kind of limited local search in the surrounding search space of the parents.

In some cases the use of the regular and specialized genetic operators is not sufficient to guarantee a proper fine-tuning of the conformation. This reflects the general knowledge that genetic algorithms are efficient for global search but do not display the same performance for local search. As a consequence, a number of different methods for local search have been proposed for the PFP. Many of such implementations are considered by authors as hybrid algorithms [51] or memetic algorithms [63, 39]. Possibly, the most popular procedures are Monte Carlo-based local search that has been used to improve solutions [15, 47, 74]. More sparsely, tabu search [36] and local hill-climbing [14, 71] are employed as genetic operators.

Another different approach is due to [51], who have proposed a local search procedure as a generalized version of the 2-opt method used for combinatorial optimization problems. This procedure starts by randomly selecting two non-consecutive amino acids in the chain and make their positions fixed in the lattice. Then, all possible conformations are evaluated, keeping the connectivity of the chain in the fixed points and changing the intermediate amino acids in between. The best conformation found in the procedure is kept. Although this procedure is computationally intensive (the number of possibilities increases exponentially as the distance between fixed points increase), it is useful to find best local conformation.

12.3.4 Ant Colony Optimization

Ant Colony Optimization (ACO) is an evolutionary technique inspired on the behavior of real ants searching for food. Possibly, [65] was the first to propose the use of ACO for the PFP. Their algoritm is based on three phases: construction, local search and pheromone updating. In the first phase, ants construct a folding over the lattice starting at a random point. Next, a greedy local search procedure is done, based on a long-range mutation method created by the authors. Then, the pheromone matrix is updated by ants, using two basic mechanisms: uniform evaporation ratio, and reinforcement of local folding motifs. They also used a mechanism of normalization of the pheromone matrix to prevent stagnation of the search. They have applied the ACO to several benchmark instances of using 2D and 3D-HP models, and results were compared with heuristic methods.

[26] also developed an ACO for the PFP using the 3D-HP model. The main difference between this implementation an that of [65] is the location of the polar amino acids, the form of the heuristic function that guides ant's decisions, and how the pheromone matrix is updated. Also, this implementation does not use any local search strategy. According the author, the implementation has achieved much better results than [65] and other heuristic methods.

Another implementation of ACO for the 3D-HP PFP is [69]. The differences of this approach to others is the use of a rapid coordinate transfer system to reduce computing time, as well a greedy local search procedure based on elementary moves, similar to the mutation operators proposed in [15, 68]. They also have devised a new method, inspired by Ethernet communication, for avoiding invalid foldings when an ant constructs a path in the lattice.

[13] has implemented single and multiple colony approaches of the ACO algorithm, with centralized and distributed processing. The main emphasis of the work was on distributed processing of multiple colonies, and they devised several methods for sharing information between evolving colonies. The several versions were tested with benchmarks of the 2D and 3D HP models. They have shown that the distributed multiple colony approach is scalable and has better performance over single colony approaches.

12.3.5 Differential Evolution

To date, the only work using Differential Evolution (DE) for the PFP is [8], using the 2D-HP model. Possibly, this is due to the fact that DE is a relatively recent evolutionary algorithm, and has been invented for continuous optimization problems. DE represents a possible solution for a problem using a vector of real numbers. The central idea of the DE algorithm is the use of difference vectors for generating perturbations in a population of vectors. This algorithm is conceptually simple, has few parameters do be tuned and, most times, converges fast to a good solution. 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 to the original DE. However, for the PFP, authors devised an adaptation to represent possible solutions to the PFP by establishing 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. They also used special strategies for mixing vectors in DE and for initializing the population. Authors applied the proposed DE algorithm to benchmark instances up to 85 amino acids and reported consistent results better than genetic algorithms and other heuristic methods. Overall, the DE approach seems to be a promising option for finding good and fast solutions for the PFP.

12.3.6 Other Evolutionary Computation Methods

Only recently that the PFP has driven the attention of researchers of the Artificial Immune Systems (AIS) area. An AIS for 2D and 3D versions of the PFP using the lattice HP model was proposed by [18, 19]. In this work, they used two entities: antigens and B cells. The search space of the problem was efficiently partitioned by memory B cells with longer life span. Another work is due to [2] who proposed an AIS hybridized with tabu search and a fuzzy inference system. A fuzzy aging operator was introduced to decide which antibodies will be deleted from the population after the selection procedure. Also, they defined a mechanism of intensive affinity maturation that uses tabu search. The proposed AIS was tested with instances of the 3D-HP model.

A hybrid approach using operators from AIS and Pareto Archived Evolutionary Strategy was used by [18] for the PFP with an all-atom model. They have compared this approach with other evolutionary computation methods when applied to a set of small proteins up to 68 amino acids.

[28] has used Evolution strategies (ES) for a sub-problem of PFP: the side-chain packing problem. They used an all-atoms representation of the backbone plus the carbon atom of the side-chain that is bonded with the central C_α. They used as energy function a measure of the deviation from a known structure. The encoding used was an array of integers representing the torsion angles for each amino acid of the chain. The evolutionary model used was a $(\mu + \lambda_t)$-ES, where μ parents generate λ_t offsprings that compete with parents for survival.

12.3.7 Other Methods

There are several implementations of different neural network architectures for the PFP. For instance, [76] uses a self-organizing map (SOM) and the 2D-HP model. However, they obtained good results only for very small sequences, up to 36 amino acids. More traditional methods, such as the well-known branch-and-bound, were applied by [12] to a benchmark of sequences of up to 100 amino acids, using the 2D-HP model. Reported results were promising, but still lacks scalability.

12.4 Open Questions

12.4.1 Models and Implementations

As a matter of fact, the most studied models for protein folding are quite distant from reality, in special the HP model. However, as mentioned before, there are still no algorithm to solve this problem in polynomially-bounded time using simple lattice models. The more complex the models, certainly, the more difficult it will be to find an efficient computational algorithm for solving the PFP. This fact suggests that there is many room for development of models that, at the same time, have more realistic features and are computationally efficient. Possibly, both hybrid and evolutionary computation methods will be of great importance in this scenery.

Two basic issues come up when observing the implementations of evolutionary computation methods for PFP, as follows:

First, the way amino acids are encoded as a possible solution may be a serious drawback. If the encoding allows invalid conformations, the search space in which the evolutionary algorithm will look for solutions will have a large amount of invalid sites. Procedures for dealing with invalid conformations may be useful. However, a more efficient search could be done if the encoding itself did not allow invalid conformations. If so, the search space could be strongly reduced and then evolutionary (or other non-exact algorithms) could be more effective in searching for the optimal conformation. Also, with the current encoding methods it is possible that a very small change in a gene (that represents, for instance, a given move in the lattice) will cause a strong change in the conformation, thus indirectly affecting the role of other genes in the encoding. This effect is known as epistasis. Those drawbacks suggest that more studies are still necessary for finding less epistatic and intrinsically collision-free encodings.

Second, the fitness function dictates the fitness landscape, that is, the shape of the search space. Most models use the number of H-H contacts as the core of the fitness function. Consequently, the corresponding fitness landscape has many discontinuities and plateaus. The first is when the number of H-H contacts vary from one conformation to the next one, and the latter, when the number of H-H contacts is the same for many different neighbor conformations (possibly, the difference between these conformations is the position of amino acids that does not account for the number of H-H contacts). Due to the embedding in the lattice, a given conformation can be rotated and/or mirrored. The same holds for portions of the conformation that are not affected by the remaining amino acids. As a consequence, it is possible to have a lot of conformations, very different each other, that have the same number of H-H contacts.

The above-mentioned facts increase the difficulty of the PFP, thus leading to an increasing loss of performance of evolutionary computation methods, as they advance towards more realistic models and protein sizes.

12.4.2 Computational Power

The NP-hardness of the PFP with lattice models was one of the main motivations for using evolutionary computation, and other heuristic methods. To date, most works have approached only small sequences, usually chains with less than 100 amino acids. It is clearly observable that evolutionary computation methods display a decreasing performance as the number of amino acids increase. On the other hand, real-world proteins have an average of 300 amino acids, and some can have thousands. Since the number of possible solutions to the PFP tend to increase exponentially as the number of amino acids increase, the use of evolutionary computation methods for larger proteins seems to be unfeasible.

In the same way, simulations using all-atoms models (or some simplified version) have been done using very small chains, far away from real-world proteins.

Apart from the intrinsic loss of performance of evolutionary algorithms for the PFP, the main factor that has set bounds on their possible performances is the available computational power. Although the memory capacity and processing speed of modern desktop computers have increased extraordinarily in last years, they are still limited for large instances of the PFP. As a consequence, recent works have reported the use of distributed/grid computing [13, 21, 49, 71] or hardware-based techniques [1] for circumventing the computational power limitation. These seem to be the direction for future research to achieve the scalability necessary for studying the folding of real-world proteins.

12.4.3 Benchmarks

All the evolutionary computation methods proposed for the PFP use a kind of supervised learning procedure. In general, a set of amino acid sequences is used as training/test cases. The results of the algorithms are compared with some previous known results, regarding the free-energy of the conformation, the compactness of the structure, the processing time, etc.

Since the lattice HP models are the most widely studied, it can be found in the literature some sets of synthetically constructed amino acid chains (not real-world proteins) ranging from 20 to 100 elements for 2D-HP and up to 64 elements for 3D-HP [38, 40, 47, 59, 74]. For more realistic models, such as those that use all-atoms approach, biological data of short length has been used, provided the tri-dimensional structure is previously known.

However, there is a large gap between the available synthetic benchmarks and real-world proteins (this is especially true for the HP models). Even considering the limited representativeness of the model, synthetic instances do not capture important peculiarities of real-world proteins. Only recently, more realistic benchmarks were proposed, based on the translation of real-world proteins to the HP model [51, 65]. There are some issues to be solved regarding the translation procedure to construct such benchmarks, and they still do not have information about the native conformation, such as minimum free-energy and tri-dimensional structure. Notwithstanding, these benchmarks represent an important improvement for this research area, offering new challenges to the existing algorithms and methods.

12.5 Conclusion

Despite the progress done using evolutionary algorithms for protein folding prediction, this is still an open problem. To date, no technique has demonstrated acceptable scalability and accuracy for problem sizes comparable to those of real-world proteins. Notwithstanding, evolutionary computation methods have been intensively used for the PSP and are the most promising.

As mentioned in Sec. 12.4, currently, there are some important questions to be addressed in PFP. The most widely used models are far from reality, but, even so, computationally complex. Further research is necessary for inventing more adequate models, encodings and fitness functions for evolutionary computation methods.

Regarding the evolutionary computation methods themselves, it seems that genetic algorithms have achieved their limit of performance. More recent evolutionary computation methods, such as AIS, ACO and DE, seem to be more promising. However, the observation of the most successful evolutionary computation methods for PFP are those that use some kind of hybridism, mainly as a local search technique, and, certainly, this is a future trend.

Another important issue to be addressed is scalability, as research moves towards realistic models and the analysis of real-world proteins. The performance of computational systems for the PFP have to increase, at least, two orders of magnitude so as to deal efficiently with real-world problems. Therefore, future trends include distributed/grid processing and specialized hardware-based approaches.

References

1. Armstrong Jr., N.B., Lopes, H.S., Lima, C.R.E.: Reconfigurable Computing for Accelerating Protein Folding Simulations. In: Diniz, P.C., et al. (eds.) ARCS 2007. LNCS, vol. 4419, pp. 314–325. Springer, Heidelberg (2007)
2. Almeida, C.P., Gonçalves, R.A., Delgado, M.R.B.S.: A Hybrid Immune-Based System for the Protein Folding Problem. In: Cotta, C., van Hemert, J. (eds.) EvoCOP 2007. LNCS, vol. 4446, pp. 13–24. Springer, Heidelberg (2007)
3. Atkins, J., Hart, W.E.: Algorithmica, 279–294 (1999)
4. Avbelj, F., Moult, J., Kitson, D.H., James, M.N.G., Hagler, A.T.: Biochemistry 29, 8658–8676 (1990)
5. Backofen, R., Will, S., Bauer, E.: Bioinformatics 15(3), 234–242 (1999)
6. Berger, B., Leighton, F.T.: J. Comput. Biol. 5, 27–40 (1998)
7. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: Nucl. Acids Res. 28, 235–242 (2000)
8. Bitello, R., Lopes, H.S.: A differential evolution approach for protein folding. In: Proc. IEEE Symp. on Computational Intelligence in Bioinformatics and Computational Biology, pp. 1–5 (2006)
9. Bornberg-Bauer, E.: In: Proc. 1st Ann. Int. Conf. on Computational Molecular Biology, pp. 47–55 (1997)
10. Chandru, V., Dattasharma, A., Kumar, V.S.A.: Discrete Appl. Math. 127, 145–161 (2003)

11. Chen, H., Zhou, X., Zhong-Can, O.-Y.: Phys. Rev. E 64, 041905–041910 (2001)
12. Chen, M., Huang, W.Q.: Genomics Proteomics Bioinformatics 3(4), 225–230 (2005)
13. Chu, D., Till, M., Zomaya, A.: Parallel ant colony optimizaiton for 3D protein structure prediction using the HP lattice model. In: Proc. 19th IEEE Int. Parallel and Distributed Processing Symp., pp. 193–199 (2005)
14. Cooper, L.R., Corne, D.W., Crabbe, M.J.C.: Comput. Biol. Chem. 27, 575–580 (2003)
15. Cox, G.A., Mortimer-Jones, T.V., Taylor, R.P., Johnston, R.L.: Theor. Chem. Acc. 112, 163–178 (2004)
16. Crescenzi, P., Goldman, D., Papadimitriou, C., Piccolboni, A., Yannakakis, M.: J. Comput. Biol. 5, 423–465 (1998)
17. Custódio, F.L., Barbosa, H.J.C., Dardenne, L.E.: Genet. Mol. Biol. 27(4), 611–615 (2004)
18. Cutello, V., Nicosia, G., Narzisi, G.: A Class of Pareto Archived Evolution Strategy Algorithms Using Immune Inspired Operators for Ab-Initio Protein Structure Prediction. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 54–63. Springer, Heidelberg (2005)
19. Cutello, V., Nicosia, G., Pavone, M., Timmis, J.: IEEE T. Evol. Comput. 11(1), 101–117 (2007)
20. Dandekar, T., Argos, P.: J. Mol. Biol. 256, 645–660 (1996)
21. Day, R.O., Lamont, G.B., Pachter, R.: Protein structure prediction by applying an evolutionary algorithm. In: Proc. 2nd Int. Parallel and Distributed Processing Symp., pp. 155–162 (2003)
22. Dill, K.A., Bromberg, S., Yue, K., Fiebig, K.M., Yee, D.P., Thomas, P.D., Chan, H.S.: Protein Sci. 4, 561–602 (1995)
23. Dinner, A.R., Sali, A., Smith, L.J., Dobson, C.M., Karplus, M.: Trends Biochem. Sci. 25, 331–339 (2000)
24. Dobson, C.M., Karplus, M.: Curr. Opin. Struct. Biol. 9, 92–101 (1999)
25. Duan, Y., Kollman, P.A.: IBM Syst. J. 40, 297–309 (2001)
26. Fidanova, S.: 3D HP protein folding using ant algorithm. In: Proc. BioPS, pp. III.19–III.26 (2006)
27. Fraenkel, A.S.: Bull. Math. Biol. 55, 1199–1210 (1993)
28. Greenwood, G.W., Shin, J.M., Lee, B., Fogel, G.B.: A survey of recent work on evolutionary computation approaches to the protein folding problem. In: Proc. Congress on Evolutionary Computation, pp. 488–495 (1999)
29. Hardin, C., Pogorelov, T.V., Luthey-Schulten, Z.: Curr. Opin. Struct. Biol. 12, 176–181 (2002)
30. Hart, W.E., Istrail, S.: J. Comput. Biol. 3, 53–96 (1996)
31. Hart, W.E., Istrail, S.: J. Comput. Biol. 4(3), 241–259 (1997)
32. Heun, V.: Discrete Appl. Math. 127, 163–177 (2003)
33. Honig, B., Cohen, F.E.: Fold Des. 1, R17–R20 (1996)
34. Honig, B.: J. Mol. Biol. 293, 283–293 (1999)
35. Hoque, M.T., Chetty, M., Dooley, L.S.: A guided genetic algorithm for protein folding prediction using 3D hydrophobic-hydrophilic model. In: Proc. IEEE Congr. on Evolutionary Computation, pp. 2339–2346 (2006)
36. Jiang, T., Cui, Q., Shi, G., Ma, S.: J. Chem. Phys. 119, 4592–4596 (2003)
37. Jiang, M., Zhu, B.: J. Bioinform. Comput. Biol. 3(1), 19–34 (2005)
38. König, R., Dandekar, T.: Biosystems 50, 17–25 (1999)

39. Burke, E.K., Krasnogor, N., Blackburne, B.P., Hirst, J.D.: Multimeme Algorithms for Protein Structure Prediction. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 769–778. Springer, Heidelberg (2002)
40. Krasnogor, N., Hart, W.E., Smith, J., Pelta, D.A.: Protein structure prediction with evolutionary algorithms. In: Proc. Int. Genetic and Evolutionary Computation Conf., pp. 1596–1601 (1999)
41. Krasnogor, N., Pelta, D., Lopez, P.E.M., Canal, E.: Genetic algorithm for the protein folding problem: a critical view. In: Proc. of Engineering of Intelligent Systems, pp. 353–360 (1998)
42. Lau, K., Dill, K.A.: Macromolecules 22, 3986–3997 (1989)
43. Lee, M.R., Duan, Y., Kollman, P.A.: J. Mol. Graph Model 19, 146–149 (2001)
44. Li, H., Helling, R., Tang, C., Wigreen, N.: Science 273, pp. 666–669 (1996)
45. Li, H., Tang, C., Wingreen, N.S.: Phys. Rev. Lett. 79, 765–768 (1997)
46. Li, Z., Zhang, X., Chen, L.: Appl. Bioinformatics 4(2), 105–116 (2005)
47. Liang, F., Wong, W.H.: J. Chem. Phys. 115(7), 3374–3380 (2001)
48. Liu, H.G., Tang, L.H.: Phys. Rev. E Stat Nonlin Soft Matter Phys. 74(5 Pt 1), 051918 (2006)
49. Liu, W., Schimidt, B.: Mapping of genetic algorithms for protein folding onto computational grids. In: Proc. IEEE Region 10 TENCON Ann. Conf., pp. 1–6 (2005)
50. Lopes, H.S., Scapin, M.P.: An Enhanced Genetic Algorithm for Protein Structure Prediction Using the 2D Hydrophobic-Polar Model. In: Talbi, E.-G., Liardet, P., Collet, P., Lutton, E., Schoenauer, M. (eds.) EA 2005. LNCS, vol. 3871, pp. 238–246. Springer, Heidelberg (2006)
51. Lopes, H.S., Scapin, M.P.: A hybrid genetic algorithm for the protein folding problem using the 2D-HP lattice model. In: Yang, A. (ed.) Success in Evolutionary Computation, Springer, Heidelberg (2007)
52. Lyngsø, R.B., Pedersen, C.N.S.: Protein folding in the 2D HP model. Technical Report RS-99-16, BRICS Bioinformatics Research Center, University of Aarhus (1999)
53. Mauri, G., Pavesi, G., Piccolboni, A.: Approximation algorithms for protein folding prediction. In: Proc. 10^{th} Ann. Symp. on Discrete Algorithms, pp. 945–946 (1999)
54. Newman, A.: A new algorithm for protein folding in the HP model. In: Proc. 13^{th} Ann. Symp. on Discrete Algorithms, pp. 876–884 (2002)
55. Nayak, A., Sinclair, A., Zwick, U.: Spatial codes and the hardness of string folding problems. In: Proc. 9^{th} Ann. Symp. on Discrete Algorithms, pp. 639–648 (1998)
56. Ngo, J.T., Marks, J.: Protein Eng. 5, 313–321 (1992)
57. Ngo, J.T., Marks, J., Karplus, M.: Computational complexity, protein structure prediction, and the Levinthal paradox. In: Merz Junior, K., LeGrand, S. (eds.) The Protein folding problem and terciary structure prediction. Birkhäuser, Boston (1994)
58. Osguthorpe, D.J.: Curr. Opin. Struct. Biol. 10, 146–152 (2000)
59. Patton, A.L., Punch III, W.F.: Goodman (eds) A standard GA approach to native protein conformation prediction. In: Proc. 6^{th} Int. Conf. on Genetic Algorithms, pp. 574–581 (1995)
60. Pedersen, C.N.S., Moult, J.: J. Mol. Biol. 269, 240–259 (1997)
61. Pedersen, C.N.S.: Algorithms in computational biology. PhD Thesis, Department of Computer Science. University of Aarhus, Denmark (2000)

62. Piccolboni, A., Mauri, G.: Application of evolutionary algorithms to protein folding prediction. In: Selected Papers from the 3^{rd} European Conference on Artificial Evolution, pp. 123–136 (1998)
63. Santos, E.E., Santos Jr., E.: Reducing the computational load of energy evaluations for protein folding. In: Proc. 4^{th} Symp. on Bioinformatics and Bioingineering, pp. 79–86 (2004)
64. Shakhnovich, E.I., Gutin, A.M.: Proc. Natl. Acad. Sci. USA 90, 7195–7199 (1993)
65. Shmygelska, A., Hoos, H.H.: BMC Bioinformatics 6, 30–52 (2005)
66. Shulze-Kremer, S., Tiedemann, U.: Parameterizing genetic algorithms for protein folding simulation. In: Proc. 27^{th} Ann. Hawaii Int. Conf. on System Sciences, pp. 345–354 (1994)
67. Socci, N.D., Onuchic, J.N.: J. Chen. Phys. 101, 1519–1528 (1994)
68. Song, J., Cheng, J., Zheng, T., Mao, J.: A novel genetic algorithm for HP model protein folding. In: Proc. 6^{th} IEEE Int. Conf. on Parallel and Distributed Computing, Applications and Technology, pp. 935–937 (2005)
69. Song, J., Cheng, J., Zheng, T.: Protein 3D HP model folding simulation based on ACO. In: Proc. 6^{th} Int. Conf. on Intelligent Systems Design and Applications, vol. 1, pp. 410–415 (2006)
70. Tang, C.: Physica. A 288, 31–48 (2000)
71. Tantar, A.-A., Melab, N., Talbi, E.-G., Parent, B., Horvath, D.: Future Gen. Comput. Syst. 23(3), 398-409 (2007)
72. Thomas, P.D., Dill, K.A.: Protein Sci. 2, 2050–2065 (1993)
73. Unger, R., Moult, J.: Bull Math. Biol. 55, 1183–1198 (1993b)
74. Unger, R., Moult, J.: J. Mol. Biol. 231, 75–81 (1993c)
75. Unger, R., Moult, J.: On the applicability of genetic algorithms to protein folding. In: 26^{th} Hawaii International Conference on System Sciences, vol. 1, pp. 715–725 (1993d)
76. Yanikoglu, B., Erman, B.: J. Comput. Biol. 9(4), 613–620 (2002)