Three-dimensional protein structure prediction based on memetic algorithms

Leonardo de Lima Corrêa[a], Bruno Borguesan[a], Mathias J. Krause[b], Márcio Dorn[a,*]

[a] *Institute of Informatics (INF), Federal University of Rio Grande do Sul (UFRGS), Av. Bento Gonçalves 9500, Porto Alegre, Rio Grande do Sul, Brazil*
[b] *Institute for Mechanical Process Engineering and Mechanics (MVM), Institute for Applied and Numerical Mathematics (IANM), Karlsruhe Institute of Technology (KIT), Karlsruhe 76131, Germany*

Keywords:
Metaheuristics
Hybrid evolutionary algorithms
Multimodal optimization
Structural Bioinformatics
PSP problem

ABSTRACT

Tertiary protein structure prediction is a challenging problem in Structural Bioinformatics and is classified according to the computational complexity theory as a *NP-hard* problem. In this paper, we proposed a first-principle method that makes use of *a priori* information about known protein structures to tackle the three-dimensional protein structure prediction problem. We do so by designing a multimodal memetic algorithm that uses an evolutionary approach with a ternary tree-structured population allied to a local search strategy. The method has been developed based on an incremental approach using the combination of promising evolutionary components to address the concerned multimodal problem. Three memetic algorithms focused on the problem are proposed. The first one modifies a basic version of a memetic algorithm by introducing modified global search operators. The second uses a different population structure for the memetic algorithm. And finally, the last algorithm consists of the integration of global operators and multimodal strategies to deal with the inherent multimodality of the protein structure prediction problem. The implementations take advantage of structural knowledge stored in the *Protein Data Bank* to guide the exploiting and restrict the protein conformational search space. Predicted three-dimensional protein structures were analyzed regarding root mean square deviation and the global distance total score test. Obtained results for the three versions outperformed the basic version of the memetic algorithm. The third algorithm overcomes the results of the previous two, demonstrating the importance of adapting the method to deal with the complexities of the problem. In addition, the achieved results are topologically compatible with the experimental correspondent, confirming the promising performance of our approach.

1. Introduction

The prediction of the three-dimensional (3-D) structure of proteins or polypeptides is one of the most important and challenging problems in Structural Bioinformatics (Dorn et al., 2014). Each protein is defined by a unique sequence of chained amino acids that under some physiological conditions fold into a particular 3-D shape (Anfinsen, 1973). The folding of an amino acid sequence is further constrained by several types of non-covalent bonds originated by interactions between different parts of the amino acid chain. These forces involve atoms in the polypeptide backbone as well as atoms located in the amino acid side-chains.

It is well known that proteins are present in all living systems, performing a variety of fundamental functions. The nature of a function performed by a certain protein is strictly related to its adopted conformation or folding. Thus, knowing the 3-D spatial arrangements of protein structures allows one to understand in a more clear way the roles performed by proteins in the cell. This is one of the primary motivations for researchers in the field. Furthermore, there is an enormous gap between the volume of data generated by the Genome Projects (Consortium et al., 2015) and the number of 3-D protein structures which are currently known and stored in the Protein Data Bank (PDB) (Berman et al., 2000). This discrepancy has motivated the development of several computational methods for the 3-D Protein Structure Prediction (PSP) problem. Hence, less than $\approx 1\%$ of non-redundant protein sequences stored in the NCBI Reference Sequence Database (Pruitt et al., 2007) have non-redundant representatives on the PDB. The PSP is a highly hard problem and has challenged Biochemists, Biologists, Computer Scientists, Physicists and Mathematicians over the last decades (Baxevanis and Ouellette, 2004; Dorn et al., 2014). The problem is classified according to the computational complexity

[*] Corresponding author.
E-mail address: mdorn@inf.ufrgs.br (M. Dorn).

theory as a NP-hard problem and describes a complicated scenario of mathematical optimization, characterized by the high dimensionality of the multimodal search space (Guyeux et al., 2014; Handl et al., 2008). The problem challenge relies on the combinatorial explosion of plausible shapes, even for a small protein, where a chain of amino acid residues ends up in a few conformations around a native state out of a vast number of possible structures (Anfinsen, 1973; Baxevanis and Ouellette, 2004).

In recent years, several computational strategies were proposed as solutions to the PSP problem. Existing methods can be categorized into four different classes (Dorn et al., 2014) according to the use of structural information from the PDB: (*i*) first principle methods without database information or *ab initio* (Osguthorpe, 2000); (*ii*) first principle methods with database information (Rohl et al., 2004); (*iii*) fold recognition methods (Bowie et al., 1991); and (*iv*) comparative modeling methods (Martí-Renom et al., 2000). Specifically, group (*ii*) represents a hybrid class of knowledge-based methods that make use of template information from experimentally determined protein structures combined with an *ab initio* approach based on simulations of physicochemical properties of the folding process in nature (Srinivasan and Rose, 1995). They do not compare the whole target protein to a known structure, but do so only for short fragments or combinations of amino acids in an attempt to get relevant information that would help in the target structure prediction. We note that our work is focused on this class of methods. For a complete description of prediction methods see Dorn et al. (2014).

In the absence of experimentally determined structures, the computational modeling of proteins can offer a suitable alternative to facilitate structure-based studies. Since PSP is a NP-hard problem, there is the need to use computational techniques that can deal with it. Metaheuristics are one of the most common and powerful techniques employed in this case. They do not always guarantee the optimal solution, but they give a good approximation with a limited computational effort (Talbi, 2009). Thus, nowadays to predict the 3-D structure of a protein, only from its linear sequence of amino acid residues, a wide range of optimization algorithms and metaheuristics are being applied (Dorn et al., 2014). However, especially in Structural Bioinformatics problems, the simple application of the canonical implementations of these methods is not enough to achieve realistic solutions. One reason for that is the severe roughness (multimodality) of the search space, mainly characterized by the several local and global minima in the energy landscape, where a small molecule can assume multiple conformations (Bryngelson et al., 1995; Handl et al., 2008).

The incorporation of previous knowledge of known protein structures stored in a protein data bank, such as the PDB, is an important strategy to improve these methods and reduce the size and complexity of the conformational search space. According to the latest editions of the *Critical Assessment of Protein Structure Prediction* (CASP), which aims to assess the current state of the art in protein structure prediction methods, the best results for the free modeling (FM) category are being achieved by knowledge-based methods (Kinch et al., 2016; Moult et al., 2016; Tai et al., 2014). Thereby, in this paper, we propose a knowledge-based computational strategy, which implements an Evolutionary Algorithm (EA) focused on the prediction of 3-D protein structures. Due to the intrinsic multimodality of the PSP problem, we aimed to incorporate concepts related to multimodal evolutionary strategies to better explore the solution space (Das et al., 2011). The discovery and maintenance of the best distinct solutions found over the optimization processes is fundamental to reveal hidden properties regarding the input target protein and reach a final set of good-enough structural models.

Our algorithm was designed to explore in a more efficient way the multimodal condition of the PSP's search space, by means of the partitioning of the state space following specific rules related to the *packing degrees* of the protein structures given by the radius of gyration (RG) measure (Lobanov et al., 2008). The main concern of the method is the maintenance of a certain diversity degree in the EA population while preserving a possible convergence state within each created solutions group. The method was combined with a local search (LS) technique to intensify the search around the most favorable regions and highlight them toward the large search space. The hybridization of global and local search techniques are commonly known as Memetic Algorithms (MAs) or Hybrid Genetic Algorithms (Moscato, 1989). MAs are based on the combination of existing algorithmic structures, avoiding the choice limitation of only one strategy to face the problem (Krasnogor and Smith, 2005; Moscato and Cotta, 2010). In many cases, the balancing between exploration and exploitation can significantly improve the search effectiveness. Nevertheless, one of the greatest challenges in MAs structuring consists in how the search space must be explored. To obtain good results through this kind of algorithm, besides an acceptable performance, it is essential to reach the correct balancing among the global and local search techniques (Boussaïd et al., 2013; Moscato and Cotta, 2010).

In this way, we have structured the presented method based on a more general Memetic Algorithm for Continuous Optimization (MACO) presented by Molina et al. (2010b), the *MA-SW-Chains* algorithm. The same idea of MA was also described in a previous work from the same authors (Molina et al., 2010a), where different LS techniques, scalability, and a set of parameters, as well as the intensity of the local searches over the global operations, were tested against a large set of continuous optimization functions. We have chosen this algorithm as it is the most recent in this line of works. It has addressed the challenge of balancing of the search space exploration inherent to the MAs, and was tested on a set of scalable optimization functions, defined in the *Special Session on Large-Scale in Global Optimization* of the *2010 IEEE Congress on Evolutionary Computation* (Tang et al., 2009). *MA-SW-Chains* has presented good results, winning the competition.

Firstly, we applied the original *MA-SW-Chains* to the PSP problem. After considering the obtained results, we modified it to better face the problem and improve results exploiting the available knowledge about the problem, developing three new algorithm versions based on the general MA. Hence, this set of new algorithms was designed by an incremental development approach. The first version of our method, the *Mod-MA*, suffered modifications only on the global operators, such as *crossover* and *mutation*. Attempting to better explore the complex search space of the problem and consider a different structured population scheme in the MA, we developed the second version. The *TT-MA* algorithm implements a structured ternary tree population of agents based on the *meme* concept (Dawkins, 1976), besides the global search operators and LS strategies already included in the *Mod-MA*. The concept of *meme* comes from the cultural evolution, and it is described as a component of cultural transmission, where complex ideas are divided into agents that propagate and mutate them while trying to keep a reasonable diversity. Each agent represents a subset of the solutions population, where the interactions between agents through global search operators and local refinements lead to the evolution and progressive improvements of the entire population. Also, in cultural evolution, ideas represent the results of search operators, and such as in culture, good ideas tend to survive while weak ones will disappear over the generations, culminating in a final set of acceptable solutions (Krasnogor and Smith, 2005; Ong et al., 2010). Similar ideas to this population scheme and the global search operators used in the designed versions were already presented in a previous work by Corrêa et al. (2016). Our aim in this work was to implement it as an incremental approach by using different components starting from the *MA-SW-Chains* algorithm together with the ones described in Corrêa et al. (2016). Therefore, the last version of the proposed method consists of an adaptation to deal with the multimodality issues of the problem. We note that none of the previously described methods have addressed such multimodality. So, the conformational search space for a given target protein is split out into different chunks from a *min-max* preset RG interval, the T-MA *TT-MA* algorithm was adapted to work with the search space break, where the protein models generated along the optimization processes are classified into the different chunks based on the RG values in order to cluster the most similar structures and keep a certain degree of population diversity. This version is referenced as *TT-MMMA* and uses the structured ternary tree population of agents to distribute the solutions over the RG intervals to facilitate the control of diversity generation and maintenance.

The most significant contribution of this work is the design and assessment of efficient evolutionary strategies and components to tackle the PSP multimodal problem. This paper is organized as follows. Section 2 presents fundamental concepts of proteins, conformational preferences of amino acids and flexibility. Section 3 shows related protein structure prediction methods. Section 4 describes the proposed methods and strategies used to deal with the PSP. Section 5 shows the computational experiments and discussion of the obtained results. Finally, Section 6 concludes the paper and points out future works.

2. Problem definition

The algorithms presented in this work use the same problem representation (Section 2.1), as well as the fitness function (Section 2.2) and the *Angle Probability List* approach (Section 2.3) described below. The algorithms receive as input parameters only the amino acid sequence of the target protein and its expected secondary structure.

2.1. Structure representation

The computational representation of a 3-D protein structure is a challenging task due to the difficulty in representing the protein structure components and simulate all the factors that contribute to the native structure stability. This representation is related to the level of detail used to describe the 3-D protein structure. The higher the number of features, the higher is the capacity of representing the protein as it appears in nature. The most detailed computational representation includes all atoms of the proteins as well as the solvent molecules (*all-atom* model). Nevertheless, using all-atom models to represent proteins is computationally expensive, and thus, simplified representations are often used (Chivian et al., 2003). In an all-atom model, the atomic coordinates in the 3-D space can be represented by a single coordinate vector X in a $3N_a$-dimensional state space, where N_a is the total number of atoms in the molecule. Since X contains three coordinates (x, y, z) for each atom, we see that for real proteins ($\approx$ 50 - 500 amino acid residues), the dimension of X is in the range of about 3000 - 30,000 positions.

Another possibility is to represent the polypeptide structure using its set of dihedral angles. This representation is based on the fact that bond lengths are nearly constant in a polypeptide chain (Neumaier, 1997). A peptide is a molecule composed of two or more amino acid residues linked by a chemical bond known as *peptide bond*. Larger peptides are called polypeptides or proteins. All amino acids found in proteins have the same main structure (main chain or backbone) and differ only in the structure of the side chain. In a chain of amino acids, the peptide bond ($\mathtt{C-N}$) (Omega angle - ω) has a partially-double bond feature and tends to be planar, presenting little or no modification. The free rotation is only permitted around the bonds $\mathtt{N-C}_\alpha$ (Phi angle - ϕ) and $\mathtt{C}_\alpha\mathtt{-C}$ (Psi angle - ψ), varying from -180° to +180° under a continuous domain. These angles are the main responsibles for the conformation adopted by a protein molecule, while the stable local arrangements of amino acids in the protein form its secondary structure. Similar to the polypeptide backbone, the side chains of a protein also have dihedral angles (Chi angles - χ), and their conformation contributes to the protein structure stabilization and packing. The number of Chi angles existing in an amino acid side chain depends on its type, ranging from 0 to 4 angles, also varying from -180° to +180° under a continuous domain. Thereby, the sequence of dihedral angles of all residues of a protein defines its 3-D conformation (Hovmöller et al., 2002). Based on that, a solution representation of a protein with N_r residues can be seen as a vector of real values of size $N_r \times 7$, but considering little modifications on the Omega angles and assigning *null* values to the missing Chi angles in the amino acid side chains. In this work, the protein structure is modeled and represented only by the dihedral angles of the backbone and side chains in order to reduce the complexity of the all-atom protein representation. The use of dihedral angles has the advantage over the Cartesian model for having reduced degrees of freedom. For the backbone representation of a polypeptide with N_r amino acids, this gives rise to $3N_r$ degrees of freedom (range of 360°). Considering the varied number of Chi angles in the side chains of the N_r amino acids, we have $3N_r + (\sum_1^{N_r} |\chi_{0-4}|)$ degrees of freedom.

2.2. Fitness function

Searching methods for the PSP problem change the orientation of atoms of the protein structure to minimize an energy function (Desjarlais and Clarke, 1998), since the native structure of a protein theoretically corresponds to the global minimum of its *Gibbs free energy* (Anfinsen, 1973). To evaluate the quality of a predicted structure, we employed the *Rosetta energy function* (all-atom high-resolution strategy) implemented by the PyRosetta toolkit (Chaudhury et al., 2010). In the Rosetta scoring function more than 18 energy terms are available, and most of them are derived from knowledge-based potentials. It is noteworthy that in the last CASP assessment, Rosetta-based algorithms achieved one of the best performance when compared to other implementations (Tai et al., 2014). The function has Newtonian physics-based terms $E_{physics-based}$ (6–12 Lennard-Jones interactions (Kuhlman and Baker, 2000) and Solvation potential approximation (Lazaridis and Karplus, 2000)). The function also has an inter-atomic electrostatic interactions which is computed through a pair potential $E_{inter-electrostatic}$ (Kuhlman and Baker, 2000) and hydrogen bond potential E_{Hbonds} (Kortemme et al., 2003). These terms are combined with a set of knowledge-based potentials $E_{knowledge-based}$ (Rohl et al., 2004) and with the free energy of the amino acids in the unfolded state E_{AA}. The total energy of a protein or residue is thus the summation of all weighted terms (Eq. (1)). The weight for each term is assigned based on the *Talaris2014* energy function, which is currently the standard Rosetta function used to evaluate all-atom structural models.

$$E_{PyRosetta} = \begin{cases} E_{physics-based} + E_{inter-electrostatic} \\ +E_{Hbonds} + E_{knowledge-based} + E_{AA} \end{cases} \tag{1}$$

In addition to the default terms of Rosetta's function, we also considered as a term the *Solvent Accessible Surface Area* (*SASA*$_{term}$) with an atomic radius of 1.4Å(Richmond, 1984) to aid on the packing of the 3-D structures. The proposed algorithms receive as input parameters the primary and secondary sequences of the target protein. Then, to improve the formation of correct secondary structures (SS), we employed the *SS term* (Eq. (2)) that was also integrated into the scoring function. The procedure gives a positive reinforcement, adding a negative constant ($-const$) to the result of the term, when the corresponding SS (zp_i) of the ith amino acid (aa_i) of the structure (Ps) that is being predicted is equal to the SS (zi_i) of the same residue of the previously informed SS of the protein. On the other hand, the technique gives a negative reinforcement to the term, adding a positive constant ($+const$), when the SS of the corresponding amino acid residues are not equal. All amino acids of the protein are comparable during the evaluation of the conformation. A simplified version of the DSSP (Kabsch and Sander, 1983) algorithm implemented by the PyRosetta Toolkit was used to assign the secondary structures along the simulation. Finally, all the terms ($E_{PyRosetta}$, $SASA_{term}$, and SS_{term}) are combined, forming the final scoring function (Eq. (4)) adopted in this work. We note that this evaluation function was also used in the work by Corrêa et al. (2016).

$$SS_{term} = \sum_{aa \in Ps} V(aa_i, zp_i, zi_i) \tag{2}$$

$$V(aa, zp, zi) = \begin{cases} -const, & zp = zi \\ +const, & zp \neq zi \end{cases} \tag{3}$$

$$E_{final} = E_{PyRosetta} + SASA_{term} + SS_{term} \tag{4}$$

2.3. Angle Probability List

The proposed methods take advantage of using experimental knowledge stored in the PDB. The primary benefit of incorporating this kind of information in a heuristic algorithm is to “decrease” the PSP complexity, reducing the size of the search space and increasing the method effectiveness. To incorporate the structural information of known protein templates and determine the conformational preferences of a target amino acid, we used a modified version of the Angle Probability List (APL)[1] scheme, proposed by Borguesan et al. (2015). The APL aims to assign the angle values to the amino acid targets through analysis of the conformational preferences of these residues in known protein structures considering their secondary structures (SS). Thus, we employed the extended version of the APL designed by Corrêa et al. (2016) in an attempt to reach more precise results and to better explore the conformational preferences of amino acids. This technique also takes into account the influence that the neighborhood of amino acids has on the reference amino acid. Beyond the *original APL*, the authors designed three other types of APL: (*i*) *APL-2l* that considers the influence of the amino acid at the immediate left position and its SS; (*ii*) *APL-2r* that examines the influence of the amino acid at the immediate right position and its SS; and (*iii*) *APL-3* that considers the importance of the amino acids at left and right and their secondary structures. The database used was built from a set of 11,130 protein structures experimentally determined by X-ray diffraction with resolution ≤ 2.5 Å, R-factor $\leq 20\%$, a and stored in the PDB until December 2015. For proteins with sequence identity above 30%, only one of them was considered. Thus, a set of 5,255,768 amino acids with occupancy equal to 1 was used for further analysis. For each amino acid residue, the dihedral angles and its secondary structure information were assigned using STRIDE (Heinig and Frishman, 2004).

To handle this information, the authors have built histograms ($H_{aa,\ ss}$) of $[-180^\circ, 180^\circ] \times [-180^\circ, 180^\circ]$ cells in order to generate different combinations of amino acid (aa) residues up to a size of three amino acids (1–3 aa) and their respective secondary structures (ss), considering the neighborhood of the reference aa for combinations with length greater than 1°. We note that unlike the fragment assembly approaches (Simons et al., 1997), in the APL each aa combination is used to assign the angles only to the reference amino acid, whereas in the fragment-based approaches the angles of all amino acids that encompass the fragment are assigned. In this way, it is possible to perform the prediction of structures that do not have a template in the PDB. Each cell (i, j) of the histogram contains the number of times that a given amino acid aa (or a combination of amino acids) has a pair of torsion angles ($i \leq \phi < i+1,\ j \leq \psi < j+1$) with the secondary structure ss. To highlight the densest conformational regions, for each cell of a given histogram we add the value of the eight neighbor cells (Eq. (5)). Then, for each H' we compute the torsion Angle Probability List (APL$_{aa,\ ss}$, Eq. (6)) that represents the normalized frequency of each square. Fig. 1 illustrates the dihedral angles distribution (ϕ and ψ) for the dataset of 5,255,768 amino acids without (Fig. 1-a) and with the normalized frequencies (Fig. 1-b). This figure also shows the different APLs for the amino acid combination “FNM” with secondary structure “CCH”: (c, d, e) represent the conformational preferences of an amino acid and its respective SS without considering neighboring amino acids (*original APL*); (f) shows the conformational preferences of the reference amino acid residue “N” considering its neighboring amino acids (left “F” and right “M”) and their secondary structures (*APL-3*). (g, h, i, j) consider a neighboring-dependent pair at right (*APL-2r*) or at left (*APL-2l*). It is easy to observe that the regions with higher frequencies change based on the amino acid and secondary structure under analysis, as well as according to the influence of the amino acid neighborhood. For a complete description of this approach, we refer our web server (*NIAS-Server*)[2] (Borguesan et al., 2016) developed to analyze the conformational preferences of amino acids in proteins.

$$H'_{aa,ss}(i, j) = \sum_{r=i-1}^{i+1} \sum_{s=j-1}^{j+1} H_{aa,ss}(r, s) \tag{5}$$

$$APL_{aa,ss}(i, j) = \frac{H'_{aa,ss}(i, j)}{\sum_{\forall x,y} H'_{aa,ss}(x, y)} \tag{6}$$

We have integrated the APL to our methods to generate short combinations of amino acids (length of 1–3 aa) in an attempt to use high-quality solutions as a starting point or after a restarting procedure (see the next sections for a complete description).

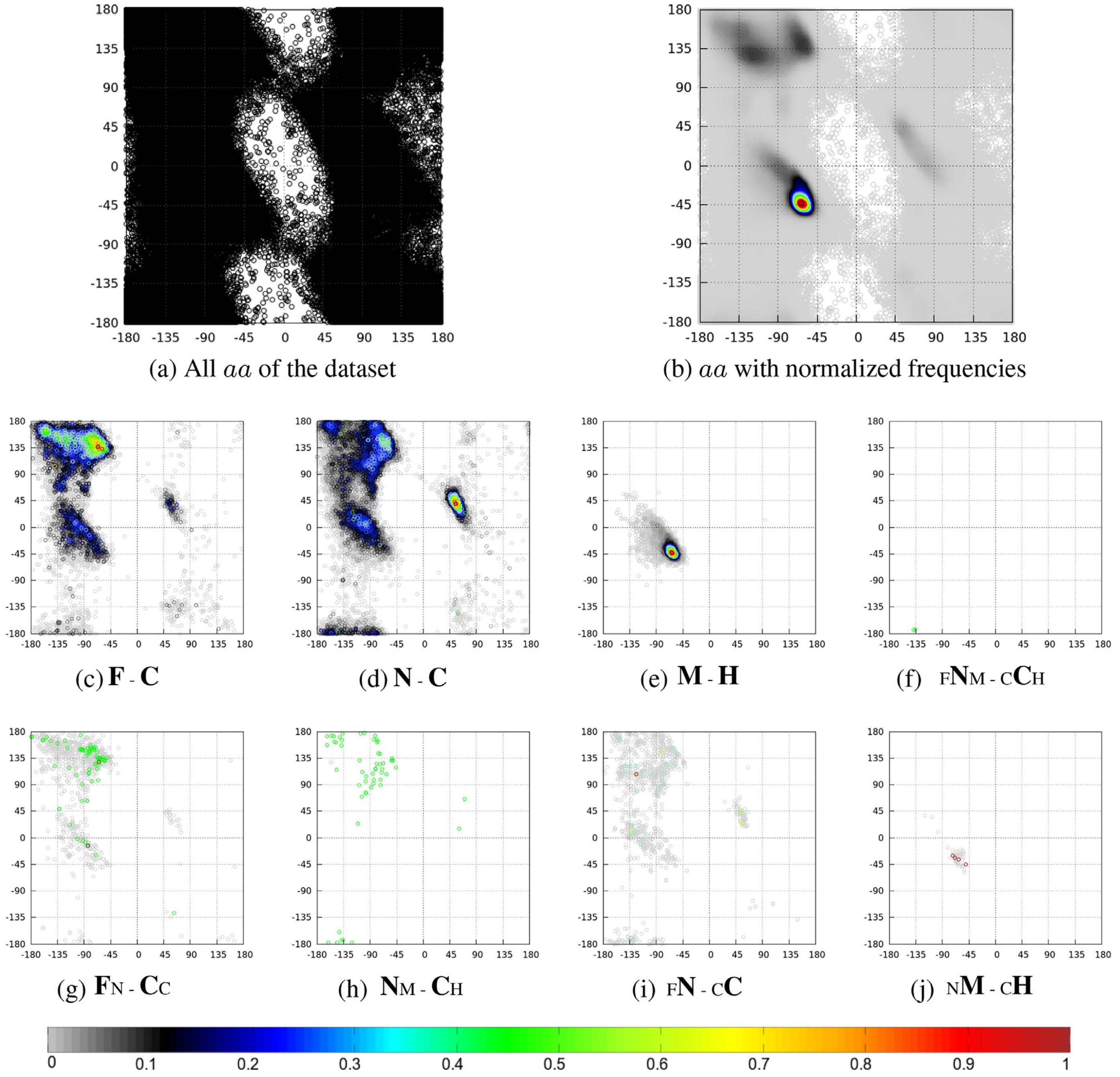
3. Related works for the PSP problem

Most of the existing challenging optimization problems cannot be optimally solved by any known computational method due to the high dimensionality and complexity of the search space (Talbi, 2009). To overcome these issues, metaheuristics techniques are being applied in an attempt to find near-optimal solutions to these problems (Boussaïd et al., 2013). Many search techniques have been proposed to deal with the PSP problem. The design of robust approaches that comprise several interconnected modules to better guide the processes by taking advantage of experimentally determined protein structures, search heuristics, screened strategies and clustering, and different types of protein representation and evaluation is being explored. For example, Elofsson et al. (1995) developed a Genetic Algorithm (GA) combined with a heuristic responsible for performing “local moves” with small modifications in the dihedral conformational space of the protein structure, to emphasize the exploitation of local minima performed by the hybrid GA. In Dorn et al. (2011), a GA with a population structured in “castes” was also allied to a path-relinking procedure (Glover, 1994) used as a Local Search strategy.

According to the latest CASP editions (Kryshtafovych et al., 2014; Moult et al., 2016), the most promising PSP methods for the

[1] http://sbcb.inf.ufrgs.br/apl.

[2] http://sbcb.inf.ufrgs.br/nias.



(a) All *aa* of the dataset

(b) *aa* with normalized frequencies

(c) **F** - **C**

(d) **N** - **C**

(e) **M** - **H**

(f) F**N**M - c**C**H

(g) **F**N - **C**C

(h) **N**M - **C**H

(i) F**N** - c**C**

(j) N**M** - c**H**

Fig. 1. Distribution of the dataset of 5,255,768 amino acids and the APL for an amino acid sequence “FNM” with secondary structure “CCH”. The dark red color marks the densest regions of the Ramachandran plot. The boldface letters represent the reference amino acids and their SS.(For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

FM category are being developed through the efforts to couple relevant structural information from known protein structures to state-of-the-art search strategies such as EAs, MAs (hybrid EAs) and replica-exchange methods. Jayaram et al. (2012) proposed the *Bhageerath* method that consists in a hybrid knowledge-based and *ab initio* model used to exploit local similarities of known proteins through Monte Carlo (MC) optimizations and smooth energy minimization techniques. In Dorn et al. (2013), a knowledge-based Genetic Algorithm was proposed aiming to reduce the size of the conformational search space taking into account previous occurrences of amino acid residues in experimentally determined proteins. This approach uses reasonable torsion angle intervals for the amino acid targets, similar to the APL idea. The use of the APL can be found in the work of Borguesan et al. (2015). In this work, the authors show the contribution of using this strategy on two different metaheuristics. Recently the authors made available the NIAS server (Borguesan et al., 2016) to compute ad-hoc APLs to take advantage of them in prediction methods or in any other problem that can use conformational preferences of amino acids. In a previously developed work by the same authors (Inostroza-Ponta et al., 2015), they have shown the first attempt of an MA that uses a variation of the APL.

Nowadays, Quark (Xu and Zhang, 2012), Zhang-Server (Zhang et al., 2016), and Baker-RosettaServer (Kim et al., 2004) can be pointed out as “reference methods” in the PSP area due to the best results achieved in the latest CASP editions. Specifically, Baker-RossetaServer is a primary web server for the Rosetta protocol (Bradley et al., 2005; Rohl et al., 2004) used to predict both *ab initio* and comparative models of protein structures (Song et al., 2013). The *ab initio* optimization of Rosetta consists in a first principle method with database information based on a fragment assembly strategy, which uses small fragments of known protein structures (3 or 9 *aa*) to generate the initial structural templates. Rosetta is divided into multiple optimization stages where different structural representations and energy functions are employed. The method starts from a low-resolution optimization, and gradually increases the level of accuracy until finalizing the process with a more precise all-atom refinement technique. Via clustering techniques and sampling of thousands of individuals, Rosetta aims to locate different conformations distributed over the search space surface. The distinct structural groups are optimized by several MC simulations, known as Replica Exchange Monte Carlo (REMC), through the processes of exchange of structural fragments. At the end of execution, the best topologically distinct structures are selected as final models.

Despite the advances in the design of computational methods for the PSP problem, the development of new strategies, the adaptation, and combination of state-of-the-art computational methods are needed. The complex energy landscape and its inherent rugosity challenge the development of more robust metaheuristics. Thus, one of the most prominent metaheuristics for solving hard optimization problems are the Memetic Algorithms. Such kind of algorithm is defined as a hybrid metaheuristic that incorporates concepts and operators of population-based global search methods, such as those present in GAs, combined with LS techniques (Moscato, 1989; Moscato and Cotta, 2010). As an example, Saleh et al. (2013) proposed an MA composed by two population-based evolutionary search strategies with coarse-grained representations and fragment assembly techniques to tackle the PSP. The authors have used two different energy functions to test the algorithm, a modified version of the *Associative Memory Hamiltonian with Water* (AMW) (Shehu et al., 2009) and the Rosetta energy function (centroid low-resolution strategy) Rohl et al. (2004). In a previous proposition by Corrêa et al. (2016), the authors developed an MA that incorporates concepts of EAs coupled with a Simulated Annealing algorithm as an LS strategy to deal with the problem. The MA comprises a ternary tree structured population of individuals, as well as ad-hoc crossover and mutation operators specifically designed for the problem, aiming to improve the quality of structural models. The authors implemented a scheme to apply the LS only in the loop regions of the individuals in an attempt to focus the search on areas that were most sensitive to the prediction. Unlike other methods, the algorithm explores the knowledge stored in PDB by using the APL strategy, to reduce the search space and to better guide the optimization (Section 2.3). Additionally, the authors proposed the evaluation function used in this work, described in Section 2.2.

3.1. Multimodal optimization

It is noteworthy that several problems of the most diverse knowledge areas encompass complex objective functions (Glibovets and Gulayeva, 2013). The energy functions used to evaluate the 3-D protein structures in the PSP, for example, fit into the complex category of multimodal objective functions (Handl et al., 2008). Structural models with similar energy values may assume very different conformations for the same target protein (Kim et al., 2009). Knowing the difficulties that the energy functions have regarding the representation of optimal points (energy values) as the best structural solutions, it is interesting to discover, throughout the execution of the optimization processes, the maximum number of distinct solutions to provide sufficient resources for future expert analysis. Thus, multimodal optimization seeks to overcome the difficulties imposed by the multimodality of the functions through adaptations in the search algorithms. The goal here is to find a varied set of solutions to the problem and not just a single one (Das et al., 2011). The discovery of multiple solutions can help on the performance of the methods since various points in the state space can be easily optimized and modified without affecting the overall processes' performance.

In this way, the EAs present advantages over other more classical search heuristics that are not population-based. Ideally, if an EA can maintain the diversity of solutions coming from an effective exploration of the search space, at the end of the algorithm execution, it is possible to obtain multiple good solutions instead of only one (Das et al., 2011). Thus, the discovery and maintenance of multiple solutions over the algorithm execution configure the main challenges in the use of evolutionary metaheuristics applied to multimodal optimization (Belda et al., 2007). The most common multimodal optimization strategies are based on the niching idea (Glibovets and Gulayeva, 2013), which is related to the attempt to find and maintain multiple groups or parts of the search space around multiple solutions in order to prevent the convergence to a single solution. Several niching methods were proposed over the years, but the central idea consists in the crowding of solutions (Thomsen, 2004) regarding some similarity criteria.

For example, in Rosetta (Rohl et al., 2004), the final result of a prediction process involves not only a single structural model, but a set of energetically favorable and topologically distinct solutions resulting from the many minimizations and clustering procedures carried out during the simulation. In the work of Garza-Fabre et al. (2016), the authors proposed an MA based on a fragment assembly technique that associates as a search heuristic the Rosetta *ab initio* protocol. As an alternative to the search space roughness, the MA uses the *stochastic ranking-based selection* procedure, which aims to minimize the evaluation function while keeping the structural diversity of the population. In addition, the method implements a modified version of the fragment-based initialization used by Rosetta in an attempt to reach an appropriate balance between the exploration and exploitation of the conformational space. Rocha et al. (2016) proposed a multi-objective GA, which uses the *phenotypic crowding* strategy as a similarity criterion for the selection of individuals. Based on this, two solutions are selected according to their structural differences. The most similar ones are selected, which implies in the delay of the population convergence. The authors also worried about the maintenance of the Pareto front diversity, incorporating the *crowding distance* technique of the Non-dominated Genetic Sorting Algorithm (NSGA-II) (Deb et al., 2002) as a criterion for the insertion of new individuals in the population. Optimizations were compared considering only a single objective against the same function decomposed into two and three objectives, and with Quark (Xu and Zhang, 2012). According to these authors, the GA was able to reach good-enough results, appearing to be promising in dealing with the PSP problem.

4. Proposed strategies

In spite of the wide range of metaheuristics proposed for multimodal and large-scale optimization, there is still the need for developing new computational methods focused on these concerns when applied to the PSP problem. Therefore, we started designing the proposed methods from a more general metaheuristic. Firstly, we used the *MA-SW-Chains* algorithm (Section 4.1) to the problem together with MA components described by Corrêa et al. (2016). We developed three new algorithm versions based on these ideas by an incremental development approach. We attempt to incorporate the PSP problem-dependencies and previous knowledge of experimentally-determined 3-D protein structures to make them more pertinent to the problem under study. We also used the APL scheme and the structural arrangements preferences of proteins (Daggett and Fersht, 2003) employed in the *APL mutation* and the *Secondary Structure Uniform crossover*. Our primary focus on this work was to propose a final method capable of dealing with the inherent multimodality of the problem by the incorporation of niching concepts, aiming at improving the state space exploration and keeping a possible trade-off between convergence and diversity of the individuals. All of these algorithms are detailed in the next sections.

4.1. MA-SW-Chains algorithm

Molina et al. (2010b) proposed a MACO, called *MA-SW-Chains*, based on the system presented in Molina et al. (2010a), combined with the *Solis and Wets* algorithm (Solis and Wets, 1981) as its LS strategy. Basically, *MA-SW-Chains* is a steady-state genetic algorithm (SSGA) plus a continuous LS technique (SSMA) that uses the concept of *LS chain* to adjust the search intensity (number of fitness evaluations) applied to the SSMA population according to the algorithm evolution. Such strategy has the objective of exploring the most promising areas of the search space maintaining the history of the LS procedures already performed on each individual. The SSMA generates only one offspring in each generation. Parents are randomly selected through a *negative assortative mating* to produce new *offsprings* by the crossover operation, which then is replaced in the population by the *standard replacement* strategy. This approach replaces an *offspring* only if it is better than the worst individual already in the population. This method was designed to produce high population diversity levels by the use of the *BLX-α crossover* (Eshelman, 1993) with a great value for its associated parameter ($\alpha = 0.5$), combined with the *BGA mutation* operator (Mühlenbein and Schlierkamp-Voosen, 1993). For every n_{frec} number of fitness evaluations of the SSMA, the local search is performed on a particular individual of the population in an attempt to improve the exploitation of global minima or escape from local minima.

– Solis and Wets Local Search: The *Solis and Wets* (SW) algorithm (Solis and Wets, 1981) consists in a randomized *hill-climber heuristic* with an adaptive step size which starts at a given point x of the energy landscape. A constant of deviate d is defined from a normal distribution with standard deviation p. If $x + d$ or $x - d$ improves the current step x, a move is performed to the better x, and *success* is recorded. Otherwise, a *failure* is recorded. The adaptive step is defined using the adjust of the parameter p according to the number of successes and failures obtained along the search. After a defined number of successes (*maxSuccesses*), p is increased to move quicker, and after a considerable number of failures (*maxFailures*), p is decreased to focus the search. Also, a bias term b is used to guide the method intended right directions. We used *maxSuccesses*=5, *maxFailures*=3, $p = 1.0$ and $b = 0$ as in Molina et al. (2010b).

– LS Chain strategy: Individuals in the population of SSMA may exist for a long time, allowing that the same individual becomes the starting point of subsequent invocations of the LS procedure. The *LS chain* strategy (Molina et al., 2010a) keeps the history of the LS parameterization of each individual to be used as the initial configuration for the next LS applications, providing an uninterrupted connection between successive LS invocations of the same individual.

– MA-SW-Chains Balancing: *MA-SW-Chains* uses a constant to regulate the *LS intensity* (I_{str}) every time that the SW algorithm is applied. The *LS intensity* is defined using the total number of fitness evaluations allowed in one search invocation. Based on this, Molina et al. (2010b) set the ratio parameter ($r_{L/G}$) responsible for balancing the efforts spent on the global search and in the refinements of the region around the most promising areas, preventing an unnecessary LS exploitation. Hence, for every n_{frec} (Eq. (7)) number of global evaluations, the continuous LS method is applied to a specific individual (c_{LS}).

$$n_{frec} = I_{str} \frac{1 - r_{L/G}}{r_{L/G}} \tag{7}$$

Starting from the best individual of the population, c_{LS} is selected if the SW algorithm has never optimized it or if it was previously refined and obtained a fitness value improvement greater than δ_{LS}^{min} (threshold). The LS is applied to the best individual that satisfies these conditions. If none fits, the SSMA population is restarted (keeping the best solution).

– Restarting: If no individual of the SSMA is submitted to the LS, the *restarting* procedure discards the entire population of the SSMA, keeping only the best solution, and generates a new one.

– Parameterization of the MA-SW-Chains: In this work, we used the same parameterization presented in Molina et al. (2010b). The SSMA population size is 60 individuals and the APL strategy initializes all of them. The *BLX-α crossover* is used with $\alpha = 0.5$. The parameter associated with the *negative assortative mating* is set to 3. Every generation, after the selection of parents, the crossover application and the replacement step, the *MA-SW-Chains* tries to apply the *BGA mutation* to the entire SSMA population with a probability of 0.125, only excluding the best solution. For each individual, mutation is applied to each ϕ and ψ angles of the amino acid residues with a probability of 0.125. The balancing parameters were defined as $I_{str} = 1000$ and $r_{L/G} = 0.5$, consequently $n_{frec} = 1000$. δ_{LS}^{min} is set to 0 as the energy function does not have any threshold value.

4.2. Mod-MA algorithm

The first version of our method uses two new global operators focused on the problem-specific properties, the *Secondary Structure Uniform crossover* and the *APL mutation*, instead of those used in the *MA-SW-Chains*. Similar versions of them were proposed in Corrêa et al. (2016). The *Mod-MA* algorithm remains an SSMA that uses the same LS combined with the *LS chain* technique presented in the *MA-SW-Chains*. The differences among them are just in the global operators. We have used the same parameter setting of the *MA-SW-Chains* except for the new operators. This version was designed to infer how much the knowledge-based operators influence in the method effectiveness. Algorithm 1 shows the pseudocode of the *Mod-MA*.

– Secondary Structure Uniform crossover: Based on the structural arrangement preferences of proteins (Daggett and Fersht, 2003), the *Secondary Structure Uniform crossover* (Algorithm 1, line 6) was designed to favor the correct formation of secondary structures. This approach prioritizes the solutions (parents in the crossover) that have already formed the proper arrangement related to the secondary sequence input parameter. It tries to keep the similarity found so far between the secondary structures of the solutions that are being worked and the previously informed secondary sequence (input parameter) to generate good offspring with correct secondary arrangements. Similar to the *uniform crossover* idea (Syswerda, 1989), for each residue (specific positions in the vector solution), the dihedral angles ϕ, ψ and $\chi_{(0-4)}$ are taken either from parent 1 or parent 2. The same probability of 0.5 is maintained if both the secondary structures related to the individuals' residues are equal or different to the previously informed secondary sequence. If only one of them is equal to the secondary structure sequence parameter, the torsional angles corresponding to this residue are assigned to the new offspring. The incorporation of such knowledge and the use of this crossover

Algorithm 1 Pseudocode of the *Mod-MA* algorithm.

```
Require: number of energy evaluations, primary and secondary amino acid sequence
Ensure: best solution found
 1: Initialize population                //Generate initial population through the APL
 2: while stop criteria not satisfied do
 3:   repeat                                              //Global search
 4:     $par_1 \leftarrow$ NegativeAssortativeMating($population$)   //Crossover
 5:     $par_2 \leftarrow$ NegativeAssortativeMating($population$)
 6:     $offspring \leftarrow$ SSUniformCrossover($par_1, par_2$)
 7:     $population \leftarrow$ StandardReplacementStrategy($population, offspring$)
 8:     for each $individual$ in $population$ do                     //Mutation
 9:       if $random\ prob < 0.125$ then
10:         $individual \leftarrow$ APLMutation($individual$)
11:       end if
12:     end for
13:     Sort $population$
14:   until $n_{frec}$
15:   $S_{LS} \leftarrow$ verify if there is any individual to be refined by LS
16:   if $S_{LS} \neq \emptyset$ then
17:     $C_{LS} \leftarrow$ best individual of $S_{LS}$
18:     if SW($C_{LS}$) was improved then                           //Local Search
19:       replacement of the former $C_{LS}$ in population by the improved $C_{LS}$
20:       Sort $population$
21:     end if
22:   else
23:     Restart $population$
24:   end if
25: end while
```

instead of the BLX-α was necessary because an operator based only on probabilities tends to disrupt the secondary structures of parents that already formed it, generating offspring with unreasonable structures. This version maintains the approach of *standard replacement strategy* (Algorithm 1 line 7) from the *MA-SW-Chains* algorithm, that replaces an *offspring* only if it is better than the worst individual already in the population.

– APL mutation: According to the conformational preferences of amino acids in proteins, the *APL mutation* tries to mutate the residues of an individual of the MA population. This mutation occurs based on a set of angles generated by the APL scheme considering the type of the amino acid and its secondary structure, not just a random value in a determined interval as in *BGA mutation*. The problem is that the BGA range can comprise "prohibited values" that are probably wrong and will not reflect the native-like angles of the protein structure as they do not have occurrences in the APL Ramachandran plots, comprising empty areas in the histograms. Thus, the routine avoids the use of angle values that do not have previously occurred in known 3-D protein structures. Similar to the *MA-SW-Chains*, every generation the *Mod-MA* algorithm attempts to apply the *APL mutation* to the entire SSMA population considering a small probability of 0.125 for each individual, excluding the best one to not worsen the solution and lose the best path found (Alg. 1, line 10). For each individual, the *APL mutation* is applied to each residue of the primary sequence also respecting a probability of 0.125. The algorithm jumps to another cell in the APL and assigns new values for the ϕ and ψ angles related to the concerned residue only if the maximum absolute difference between the current and new angles (ϕ and ψ) is not greater than the *jump* parameter (diversity control). We fixed the constant $jump = 50$.

4.3. TT-MA algorithm

To better explore the complex search space of the problem and consider a new structured population scheme in the MA, we have designed the second version of our method. *TT-MA* is an MA that uses a structured ternary tree population (Fig. 2) instead of the SSGA, combined with the specifically designed global search operators already incorporated in the *Mod-MA* (Section 4.2). The organization of the MA population in a ternary tree was presented in the previous work by Corrêa et al. (2016) to tackle the PSP problem. Each node of the tree represents an agent that stores a subset of solutions. All of the agents' solutions form the entire population of the method. As in the *Mod-MA*, this approach also uses the SW algorithm and the *LS chain* strategy, which differs from the *Simulated Annealing* algorithm (Kirkpatrick et al., 1983) used as LS in Corrêa et al. (2016). The interactions between agents give the optimization of solutions through global searches and local refinements, which leads to the evolution and progressive improvements of the entire population. Basically, *TT-MA* assembles all of the components already described with a different organization of the population. With this, we can assess the role of the ternary tree in the MA performance when compared with the previous versions.

– Population definitions: The population of the *TT-MA* is composed of thirteen agents organized in a hierarchical ternary tree that forms four overlapped subpopulations consisting of three supporters and one leader agent. Each agent maintains a set of n solutions where one of them is called *current solution* and the others are the *pocket solutions* (Fig. 2). In this work, we adopted 6 solutions per agent to be compatible with the previous algorithm versions. The agents can only interact with the "leader agent" of the subpopulation to which they belong. The pocket solutions are the best solutions found so far, and the current solution represents the one that is being modified in the current generation of the algorithm. Algorithm 2 shows the pseudo-code of the *TT-MA*.

– Interactions between agents: according to the Algorithm 2, in each generation of the global search step, the "leader agent" of a subpopulation applies the *Secondary Structure Uniform crossover* on the agents located in the lower level of its subpopulation (Algorithm 2, line 8). The parents for the crossover are randomly selected from the pocket solutions of the concerned agents (Algorithm 2, lines 6 to 7). Then, the generated offsprings are

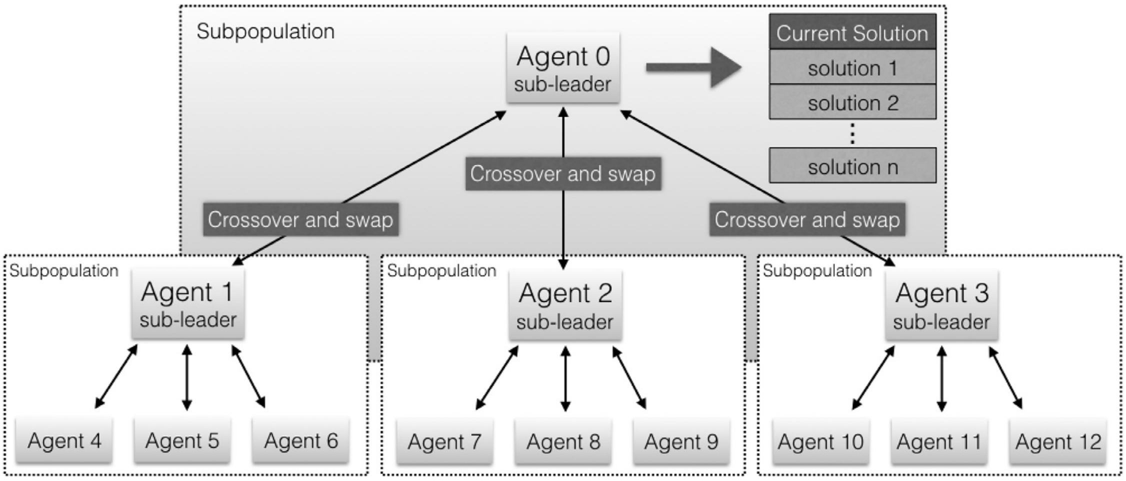


Fig. 2. Structured ternary tree population used in the MA. On the top right, it illustrates an agent composed by n solutions: a current one and the pocket solutions. Adapted from Corrêa et al. (2016).

Algorithm 2 Pseudocode of the *TT-MA* algorithm.

```
Require: number of energy evaluations, primary and secondary amino acid sequence
Ensure: $sol_{best}$: best solution found
 1: Initialize population                          //Generate initial population through the APL
 2: $sol_{best} \leftarrow$ best solution of $agent_0$
 3: while stop criteria not satisfied do
 4:   repeat                                                   //Global search
 5:     for each agent do                                      //Crossover and Mutation
 6:       $par_1 \leftarrow$ random solution of the sub-leader agent
 7:       $par_2 \leftarrow$ random solution of the agent
 8:       $offspring \leftarrow$ SSUniformCrossover($par_1$, $par_2$)
 9:       $agent.current \leftarrow$ APLMutation($offspring$)
10:     end for
11:     Sort population
12:     Update population
13:   until $n_{frec}$
14:   for each agent do                                        //Local search
15:     $agent\ sol_{best} \leftarrow$ SW($agent.sol_{best}$)
16:   end for
17:   Update population
18:   if best solution of $agent_0 < sol_{best}$ then
19:     $sol_{best} \leftarrow$ best solution of $agent_0$
20:   end if
21:   if No Improvement Threshold reached then
22:     Restart population
23:   end if
24: end while
```

submitted to the *APL mutation* with a probability of 0.125 and stored without restrictions in the corresponding current solutions of the supporter agents (Algorithm 2, line 9). Each agent keeps the pockets always sorted according to the best energy found so far (Algorithm 2, line 11). After a generation has been completed, the population is updated (Algorithm 2, line 12) following three steps: (*i*) the current solution of each agent is added to the pocket solutions if it is better than one that is already stored; (*ii*) in the subpopulations 1, 2 and 3, the lower level agents send their best solutions to the leader agents and receive back the worst ones, characterizing the *swap operation*; (*iii*) the supporter agents of the subpopulation "0" also perform the *swap*, sending their best solutions to the $agent_0$ and receiving back the worst ones. Therefore, the best solutions are kept on the top of the hierarchy in the pockets of the $agent_0$. As in the previously described SSMAs, every n_{frec} number of global energy computations, the SW algorithm is applied to the best solution of each agent (Algorithm 2, line 15) using the *LS Chain* technique. Still, the *restarting* procedure (Algorithm 2, line 22) is applied when the algorithm reaches a premature stabilization (Algorithm 2, line 21). If the best solution of the $agent_0$ has not been improved over three turns of global and local search execution, it will restart the population by keeping only the best-known solution of each agent. We emphasize that we kept the same parameter setting already defined in the previous algorithms.

4.4. TT-MMMA algorithm

As already stated, our central objective in this work was to design an incremental approach using the combination of promising evolutionary components to finally address the multimodal PSP problem. Thus, the last version of the MA is a variation of the *TT-MA* algorithm, described above. It incorporates concepts related to evolutionary multimodal strategies (Das et al., 2011) to better guide the solution space exploration in an attempt to discover and optimize a set of distinct structural solutions instead of spending efforts to optimize only one solution when the MA reaches a convergence state. It is noteworthy that it is not only the discovery but also the maintenance of the best distinct solutions found for the optimization processes, which are fundamental to reach a final set of different structural models. The multimodal optimization strategies try to divide in some way the state space, be it by partitioning the problem space or by creating multiple clusters around the most distinct solutions, in order to prevent the convergence of the entire population of the algorithm to a single point. This "restriction" imposed to the method can balance the computational efforts to focus on more than one promising solution. None of the previously described algorithm versions have addressed this. Since the PSP conformational space is known by the severe roughness and the enormous complexity due to the high dimensionality of variables (Bryngelson et al., 1995; Handl et al., 2008), *TT-MMMA* was developed to deal with these complexities that are inherent to the solution space.

In this way, we decided to divide the conformational search space of the problem, following a particular structural measure of proteins to maintain some level of diversity in the population of the MA and explore more distinct folds while preserving a possible convergence state in each created niche. The partitioning of the solution space was defined by specific rules related to the *packing degrees* of the protein structures given by the radius of gyration (RG) measure (Lobanov et al., 2008). The RG of a protein structure is the root mean square distance of the protein atoms from its center of mass. The RG can be used as a packing indicator, since the lower the RG, the higher the proximity of the atoms with the center of the protein. If a protein structure is stable next to its native state, the RG will probably remain stable. However, when the protein is unfolded (less stable conformation), the RG values tend to vary.

For a given target protein, its search space is split out into different chunks (sub-intervals) from a *min-max* preset RG interval. The range of the minimum and maximum RG values of this interval defines the possible values that the protein models (solutions) can assume during the optimization regarding the target protein. The *TT-MMMA* was adapted to work with the search space break. The protein models generated along the optimization processes are classified into the different chunks based on their RG values to cluster the most similar structures and keep some level of population diversity. The idea of partitioning the range of RG values into sub-intervals forces population of the method to always keep distinct solutions throughout the optimization. The RG interval thresholds are established according to specific characteristics of the target protein, which consider the length of the amino acid sequence and its structural class. The class of a protein is defined considering the structural arrangements and components of the secondary structure (SS). From the length of the target amino acid sequence and its class, the minimum and maximum thresholds of the RG interval are defined by analyzing experimental protein structures that follow the same pattern (length and class).

– Protein classes: Proteins can be classified according to their SS components and arrangements. The classification of proteins into different structural classes can provide detailed descriptions and insights about the relationships and in common features among them. The classes configure different molecular interactions, which originate different SS arrangements and 3-D topologies. In this work the proteins were classified into five classes. This classification follows the predominance values delineated in the work of Chou (1995), which comprises: (*i*) class of α-helices, covering proteins that have more than 40% of α-helices and less than 5% of β-sheets in their SS composition; (*ii*) class of β-sheets, which comprises proteins that have more than 40% of β-sheets and less than 5% of α-helices; (*iii*) class of irregular regions, which includes proteins with less than 10% of α-helices and β-sheets; (*iv*) class of α- and β-proteins, which encompass proteins that have more than 15% of α-helices and β-sheets; and (*v*) hybrid class, which comprises proteins that do not fit into any of the previous classes, i.e., they present a combination of the three types of SS in their SS composition.

– Protein database: The definition of the RG interval for a given target protein is done through the correlation between its length and structural class and the experimentally determined protein structures that follow the same pattern (length and class). To search the corresponding experimental proteins, we used the same protein database developed for the APL strategy. Its specifications were already detailed in Section 2.3 and includes 11,130 protein structures obtained from the PDB. Thus, for all of the proteins in the dataset, we have defined their structural classes and calculated the RG measure.

– Definition of the RG interval: For a certain target protein, the *min-max* RG thresholds are defined by querying the idealized database, relating the length of the amino acid sequence and its class defined by the SS. The query returns a set of proteins compatible in length and class with the target. From the returned set, the minimum and maximum thresholds are assigned from the lowest and highest RG values attached to the returned structures, respectively. We defined a minimum number of 5 proteins to define the RG interval. To ensure that at least 5 proteins are returned by the query, the length parameter is modified whenever this condition is not satisfied. If the length of the returned set is less than 5 (condition not satisfied), the parameter length is increased by ± 1. It allows proteins with length equal, greater and smaller than the target, respecting the current length parameter. So when the condition is not satisfied, a new query is performed using the modified length parameter. This procedure is repeated until a representative set is returned.

– Population definitions: As a variation of the *TT-MA* algorithm, the *TT-MMMA* keeps the same MA components already included in it. The algorithm makes use of the structured ternary tree population of agents to distribute the solutions over the chunks created by the partitioning of the RG interval to facilitate the control of diversity generation and maintenance. Given that the ternary tree (Fig. 2) has nine agents in its lowest level, the RG interval for a certain target protein is also divided into 9 chunks, such that each lower level agent (leaves) is associated with one of these. For example, the $agent_4$ is assigned to the first chunk of the interval and the $agent_{12}$ to the last one. The sub-leader agents are associated to the chunks of their children in the tree, e.g., the $agent_1$ is assigned to the first three chunks. The $agent_0$ (root) encompasses the whole RG interval. These associations mean that over the optimization process, the RG measures of the solutions of an agent must be within the range of its chunk.

To ensure the property that the RG of the solutions of an agent will be in the range of its associated RG chunk over the optimization, the interactions between agents described in the *TT-MA* had to be slightly changed. The agents still interact with the "leader agent" of the subpopulation to which they belong to, but now they can interact with the agents located in the same level of the tree (horizontal interactions). For example, $agent_4$ communicates with the other 8 leaf agents. In general, the algorithm of the *TT-MMMA* is the same as the *TT-MA*. The only differences are in the *update function* (Algorithm 2, lines 12 and 17) and in the LS technique (Algorithm 2, line 15). The value of the application threshold (Algorithm 2, line 21) of the *restarting* (Algorithm 2, line 22) was also modified.

– Interactions between agents: According to the Algorithm 2, after a generation of the global search has been completed or after the execution of the LS, the population is updated. The *update* *function* is responsible for adding the current solution of an agent to its pocket solutions and performing the *swap operation* between agents. Thus, the RG interval restriction was only imposed to the pocket solutions. The current solution can assume any RG value since it is the individual which is modified over the generations of the algorithm. To ensure that each agent stores in its pocket only solutions with RG compatible with the range of its chunk, the *update function* for the TT-MMMA was modified and follows the steps below:

1. The current solution of each agent is added to the pocket solutions if they are better than one that is already stored and if their RG value is within the range of agent's RG chunk;
2. If the current solution of an agent is not added to its pocket solutions because it is out of the range of its chunk (step 1), then the agent tries to add this solution to the pocket solutions of the other agents located in the same level of the tree (horizontal interaction), starting from its neighbors. If the solution does not fit in any RG chunk, then it is not stored. In the next generation, it will be replaced anyway (Algorithm 2, line 9);
3. In the subpopulations 1, 2 and 3, the lower level agents send their best solutions to the leader agents and receive back the worst ones, characterizing the *swap operation*. The agents only swap solutions if the solution of the leader agent fits in the RG chunk of its child and if the solution of the child fits in the chunk of the leader agent. If some of them do not fit, then the operation is not performed;
4. The supporter agents of the subpopulation 0 also perform the *swap*, sending their best solutions to the $agent_0$ and receiving back the worst ones. The operation is performed if the solution of the $agent_0$ is within the range of the chunk of the supporter and if the solution of the supporter is in the range of the chunk of the $agent_0$. If some of them do not fit, then the operation is not performed.

With these modifications in the *update function* we ensure that the solutions stored in the agents' pockets respect the ranges of the chunks associated with the agents. Since the LS strategy is applied directly to the best solutions of each agent, it was also modified to accept only moves that respect the range of its chunk. The application threshold of the restarting was increased from 3 to 10 since the division of the search space already increases the population diversity. Hence, if the best solution of the $agent_0$ has not been improved over ten turns of global and local search, it will restart the population by keeping the best-known solution of each agent. We kept the same parameter setting described in the previous algorithms.

– Adaptation to the spot: All of agents' solutions are initialized by the *APL strategy* without restrictions of RG interval. So at the beginning of the simulation or after a restarting, the solutions are not following the range of the RG chunks of the agents. The property of restriction of RG interval of the agents' solutions appears as they begin to update the population over the optimization since no solution out of the range of the chunks is inserted in the population. Thus, as the solutions start to be stored in the pocket solutions, this property emerges. We called this pattern as *adaptation to the spot*, which means that due to the restrictions imposed in the update function the solutions are gradually adapting to the chunk of the agents. At the end of the simulation, the agents will present distinct solutions with different conformations and *packing degrees*.

5. Computational experiments

All of the algorithms described in this work were coded in Python. They were run 30 times with stop criterium of 10^6 evaluations of energy per run on each target protein. Tests were performed in an Intel Xeon E5-2650V4 30 MB, 4 CPUs, 2.2Ghz, 96 cores/threads, 128G, 4TB. The sequences of sixteen small proteins ranging from 14 to 87 amino acids were obtained from the PDB and used as case studies in our experiments. These targets were selected taking into account different sizes and the secondary structure content. Table 1 presents details of the target protein sequences. We note that the knowledge of algorithms was restricted regarding the target proteins to test the algorithms as if we were performing a prediction with any similar structure in the PDB (Free Modeling category). To guarantee that the proposed method does not take advantage of any protein structure from the PDB with a high level of similarity to the targets, we removed from the APL database all of the protein structures indicated by the SAS[3] (*Sequence Annotated by Structure*). Also, to situate our methods according to the most relevant methods in the field, we have done a comparison with the Rosetta *ab initio* protocol (Rohl et al., 2004). As already mentioned, according to the latest CASP editions, Rosetta is in the state-of-the-art and is one the most promising methods to deal with the problem. The computational experiments aimed to analyze the behavior of the algorithms regarding energy and to measure the biological significance (quality) of the best solutions found. All of the described algorithm versions were compared, including the *MA-SW-Chains* and the previously proposed MA of Corrêa et al. (2016).

Table 1
Amino acid sequences used to test the proposed algorithm versions. The second column shows the number of residues, and the third column shows the secondary structure components.

PDB ID	Target length	SS Content
1ACW	29	One β-sheet/One α-helix
1CRN	46	One β-sheet/Two α-helices
1ENH	54	Three α-helices
1K43	14	One β-sheet
1L2Y	20	Two α-helices
1Q2K	31	One β-sheet/One α-helix
1ROP	63	Two α-helices
1UTG	70	Five α-helices
1WQC	26	Two α-helices
1ZDD	35	Two α-helices
2MR9	44	Three α-helices
2P5K	64	One β-sheet/Three α-helices
2P6J	52	Three α-helices
2P81	44	Two α-helices
2PMR	87	Three α-helices
3V1A	48	Two α-helices

5.1. Results and discussion

For each case study, we present a structural analysis of the solutions among the 30 performed runs. The quality of the predicted structures was evaluated by similarity comparisons with experimentally determined protein structures regarding the *root mean square deviation* (RMSD, minimization measure) (Zhang and Skolnick, 2004) and the *global distance total score test* (GDT_TS, maximization measure) (Zemla, 2003). Table 3 shows the final results of the *MA-SW-Chains* (M1) and the three proposed algorithm versions, *Mod-MA* (M2), *TT-MA* (M3) and *TT-MMMA* (M4) applied to the target proteins. It also shows the results of the MA proposed in Corrêa et al. (2016) (M5) and the Rosetta *ab initio* protocol (R.), both of them applied to the same set of target proteins. Table 2 summarizes the main components and differences among methods developed in this work.

– Comparisons between methods M1, M2, M3 and M4: Analyzing the results of the Table 3, we observe that in the average

[3] http://www.ebi.ac.uk/thornton-srv/databases/sas/.

Table 2
Variations of the proposed algorithm versions developed based on an incremental approach by means of the combination of promising evolutionary components to finally address the PSP as a multimodal problem. All of the four methods use only the experimental knowledge provided by the APL.

	Population		Crossover		Mutation		Multimodal
	SSMA	Ternary Tree	*BLX-α*	SS Uniform	BGA	APL	
M1	X		X		X		
M2	X			X		X	
M3		X		X		X	
M4		X		X		X	X

of the 30 runs, the methods M2, M3, and M4 outperformed the M1 regarding the RMSD and GDT_TS in all of the case studies. It can also be noticed in the results related to the lowest RMSD and highest GDT_TS, with some exceptions. Nevertheless, M2 and M3 did not present significant differences in the average of the cases. Both methods showed similar results; method M2 achieved better average results of RMSD and GDT_TS in 7 cases while M3 achieved better average results in 5. In the other 4 cases, M2 and M3 obtained equal results or while one achieved a better average result of RMSD, the other performed better in GDT_TS, and vice versa. M3 produced better results in 6 targets considering the lowest RMSD and 9 cases regarding the highest GDT_TS. These results show that the final structures of each method tend to be different and may point out that both versions are capable of generating better solutions than the general method M1. Probably, one reason for that is the combination of the parameter set defined in the *MA-SW-Chains* with the incorporation of previous knowledge about the PSP problem, e.g., the knowledge-based global operators. It was able to reduce the size and complexity of the conformational search space and facilitate the search. This combination aided in the good performance of the algorithms as the exploration was improved and more refined solutions were found. Thus, it is possible to state that the correct balancing (trade-off between global and local search) of the *MA-SW-Chains* was kept in the subsequent versions. Such results also reinforce the need to include previous knowledge about the problem in the search strategies. We still observe that these analyses indicate that the organization of the population in a ternary tree is not as effective as the incorporation of specific-problem properties and the correct parameterization and balancing of the MA. Although M3 did not surpass the results of M2 in its absolute majority, it was important to give rise to the multimodal adaptations implemented in M4, since the partitioning of the search space and the clustering of similar solutions according to their RG values were designed based on the ternary tree structure and the agents' interactions.

Regarding results of the method M4 summarized in the Table 3, we notice that the method outperformed all of the previous algorithms (M1, M2, and M3) concerning the average results of RMSD and GDT_TS in almost all of the targets, except for the average RMSD of the 2P81, 1ZDD and 1K43, and GDT_TS of 1K43. Similar results can be seen analyzing the lowest values of RMSD, where the method achieved better or equal results in 14 targets. For the highest values of GDT_TS, M4 obtained better results in 9 cases. We observe that the last version of the incremental MA, which comprises the promising MA components included on the previous algorithm versions combined with the multimodal strategy, was able to better guide the conformational space exploration and, consequently, find better solutions facing a multimodal and complex problem such as the PSP. M4 overcame the results of its previous versions. Thus, these results demonstrate the importance of adapting the method to deal with the multimodality issues of the problem employing the generation and maintenance of the population diversity over the optimization process.

– Execution analysis of the method M4: Fig. 3 illustrates three scenarios of the optimization processes of the M4 method for three target proteins: (*i*) the leftmost plot shows the average RG of the sub-population of each agent throughout the algorithm execution; (*ii*) the central plot shows the energy convergence curve of the best solution of each subpopulation and the average energy of all of the agents' solutions; and (*iii*) the rightmost plot shows a comparative analysis between the best solution of each subpopulation at the beginning of the optimization and at the end of it, according to the energy and RMSD values. We note that the plots (*i*) and (*ii*) in Fig. 3 show only a piece of the generations of the algorithm related to the execution that reached the lowest value of RMSD out of the 30 runs. Each generation represents a complete cycle of global and local searches, which means that according to the parameters of balancing ($I_{str} = 1000$, $n_{frec} = 1000$) in Algorithm 2, each generation represents 1000 energy evaluations both for global and local searches. The LS was applied to the best solution of each agent. The plot (*iii*) shows the initial and final individuals of the agents of the 30 runs performed. The complete description of the three scenarios for all of the target proteins is included in the supplementary material (Fig. 1-3).

From the first scenario illustrated in Fig. 3, it is possible to note that the average RG of the solutions of each agent tended to concentrate according to the range of the established chunks for each agent. The blue color of the $agent_4$ in the graph illustrates the chunk with the lower RG values while the orange color of the agent $agent_{12}$ shows the chunk with the higher RG values of the interval. The peaks in the graphs represent the restarting procedure, and with this we observe the capacity of the method to gradually adjust each sub-population to its correct range of RG values, exemplifying the pattern called as *adaptation to the spot*. We can also observe that the partitioning of the search space following the RG interval of a given target protein and the restrictions imposed to force each agent to optimize in a different RG interval ensured the generation and maintenance of a diverse set of solutions (with different *packing degrees*) over the optimization process. Fig. 2 in the supplementary material shows a comparative analysis between average RG of the sub-population of each agent (*scenario i*) over the M3 and M4 executions for all of the target proteins. This comparison reinforces the multimodal optimization capacity of M4. We see that the method M3 tends to fastly converge to a single local optimum given the low variation of the average RG of the solutions of each agent. The second scenario shows the capacity of the method to optimize distinct structural models and still converge to similar energy values. These plots illustrate the multimodality of the energy function, where structural models with similar energy values may assume different conformations for the same target protein. Corroborating with this, the scenario iii shows that the method was able to optimize the initial individuals while keeping their structural differences. We observe that the final individuals tended to finish with similar energy values but different RMSD values, which confirms the structural differences of the solutions, the roughness of the search space and

Table 3

Simulation results of the proposed methods. M1 represents the *MA-SW-Chains* algorithm, M2 is the *Mod-MA*, M3 is the *TT-MA* and M4 is the final version *TT-MMMA*. Method M5 is the MA proposed in the previous work by Corrêa et al. (2016) and the R. is the Rosetta protocol. The **boldface** numbers are the best results regarding Energy, RMSD and GDT_TS, excluding the Rosetta results. The (*) denotes the case studies where Rosetta outperformed all of the others.

ID_PDB	Energy		RMSD		GDT_TS	
	Lowest	Avg. (std)	Lowest	Avg. (std)	Highest	Avg. (std)
1ACW-M1	−8634.1	2134.6 ±(4874.1)	3.3	7.6 ±(1.9)	62.1	44.8 ±(5.9)
1ACW-M2	−13312.8	−2475.4 ±(6809.1)	3.4	6.7 ±(2.0)	64.7	51.3 ±(5.3)
1ACW-M3	−13443.8	−3825.4 ±(6615.1)	2.9	6.9 ±(1.7)	62.1	49.9 ±(4.9)
1ACW-M4	−**23152.1**	−**22872.1** ±(210.3)	1.6	**2.7** ±(1.5)	79.3	**70.0** ±(6.9)
1ACW-M5	−12400.9	−11582.8 ±(578.1)	**1.4**	3.8 ±(1.9)	**82.8**	63.5 ±(9.2)
1ACW-R.	−31.8	−25.0 ±(5.2)	1.5	*2.3 ±(0.9)	*82.8	*72.6 ±(6.3)
1CRN-M1	−7907.8	−6890.5 ±(754.7)	8.1	11.3 ±(1.7)	41.3	33.8 ±(3.8)
1CRN-M2	−11911.8	−8784.8 ±(811.5)	6.1	10.4 ±(2.3)	**60.9**	42.2 ±(5.2)
1CRN-M3	−9588.5	−8100.9 ±(870.0)	6.3	10.5 ±(2.1)	49.5	41.1 ±(3.2)
1CRN-M4	−**39923.7**	−**33723.4** ±(2215.8)	4.0	**8.4** ±(1.8)	54.9	**46.6** ±(3.5)
1CRN-M5	−12599.6	−2520.1 ±(2985.1)	**3.8**	9.2 ±(2.6)	60.3	44.1 ±(7.1)
1CRN-R.	−57.9	−43.9 ±(11.8)	*2.8	*4.8 ±(1.0)	*76.1	*63.4 ±(6.9)
1ENH-M1	−32338.8	−30603.2 ±(957.4)	3.1	13.1 ±(3.3)	45.4	32.0 ±(3.4)
1ENH-M2	−33503.0	−32838.6 ±(304.0)	3.5	8.9 ±(2.9)	42.6	37.6 ±(3.0)
1ENH-M3	−32886.5	−32322.4 ±(425.7)	3.1	10.2 ±(3.0)	**47.2**	36.6 ±(3.2)
1ENH-M4	−**49321.6**	−**48820.9** ±(175.8)	**2.1**	**6.2** ±(3.2)	46.3	**40.2** ±(3.6)
1ENH-M5	−32685.2	−32166.0 ±(388.6)	2.7	6.5 ±(2.6)	46.8	39.4 ±(3.6)
1ENH-R.	−102.1	−86.6 ±(6.6)	*1.2	*2.7 ±(1.1)	*49.5	*44.4 ±(1.6)
1K43-M1	−4182.2	−2794.0 ±(1747.2)	0.6	1.2 ±(0.6)	**89.3**	79.2 ±(6.4)
1K43-M2	−4441.0	−4281.8 ±(363.2)	**0.5**	**1.0** ±(0.2)	**89.3**	**82.0** ±(4.0)
1K43-M3	−4481.9	−4334.4 ±(63.1)	0.6	**1.0** ±(0.2)	87.5	81.4 ±(4.1)
1K43-M4	−**12589.4**	−**12525.1** ±(35.0)	**0.5**	1.1 ±(0.2)	87.5	78.5 ±(4.4)
1K43-M5	−4601.8	−4495.8 ±(61.2)	0.6	**1.0** ±(0.2)	85.7	78.9 ±(4.0)
1K43-R.	4.8	133.0 ±(242.9)	0.6	*0.9 ±(0.1)	85.7	80.0 ±(3.1)
1L2Y-M1	−6885.1	−6287.5 ±(343.1)	2.2	4.7 ±(1.0)	71.3	59.7 ±(6.6)
1L2Y-M2	−7286.2	−5357.1 ±(2614.6)	1.7	3.7 ±(1.2)	78.8	70.2 ±(4.9)
1L2Y-M3	−6367.1	−5935.7 ±(252.2)	1.3	2.7 ±(1.0)	85.0	73.4 ±(5.9)
1L2Y-M4	−**13774.0**	−**13708.1** ±(33.6)	**1.0**	2.0 ±(0.7)	**86.3**	**79.0** ±(4.7)
1L2Y-M5	−3238.9	−2307.3 ±(245.2)	1.1	**1.9** ±(0.4)	85.0	78.3 ±(4.1)
1L2Y-R.	−33.7	−26.9 ±(4.4)	*0.6	*1.4 ±(0.3)	*96.2	*82.1 ±(5.1)
1Q2K-M1	−5624.6	−870.2 ±(1962.4)	2.9	7.5 ±(2.3)	67.7	48.8 ±(8.2)
1Q2K-M2	−12119.6	−5195.4 ±(4651.5)	2.7	5.6 ±(2.1)	70.2	58.8 ±(5.8)
1Q2K-M3	−12797.7	−5132.7 ±(3948.3)	2.8	5.3 ±(1.7)	75.0	59.0 ±(5.9)
1Q2K-M4	−**28581.1**	−**25975.6** ±(2433.0)	**1.4**	**3.6** ±(0.9)	**83.1**	**65.2** ±(4.8)
1Q2K-M5	−16456.3	−13106.5 ±(2485.1)	2.0	3.8 ±(0.9)	79.8	63.5 ±(5.4)
1Q2K-R.	−39.3	−28.3 ±(8.2)	*0.6	*1.8 ±(0.8)	*97.6	*81.0 ±(9.8)
1ROP-M1	−45103.9	−44844.2 ±(171.6)	4.9	13.0 ±(3.8)	56.3	48.2 ±(4.8)
1ROP-M2	−46705.3	−46129.0 ±(298.5)	**1.8**	7.8 ±(5.1)	**81.3**	59.9 ±(9.4)
1ROP-M3	−46412.7	−45961.2 ±(268.3)	2.4	7.4 ±(4.5)	75.0	59.3 ±(7.9)
1ROP-M4	−**51715.4**	−**51496.6** ±(103.6)	**1.8**	**3.0** ±(0.7)	78.1	**69.5** ±(3.7)
1ROP-M5	−47027.1	−46683.8 ±(262.3)	1.9	3.2 ±(0.9)	76.8	67.3 ±(5.8)
1ROP-R.	−101.1	−86.1 ±(8.9)	*1.1	5.6 ±(2.9)	*88.8	61.9 ±(13.9)
1UTG-M1	−46770.3	−43663.7 ±(1455.6)	10.2	16.7 ±(3.6)	37.9	30.4 ±(3.3)
1UTG-M2	−48884.3	−47704.5 ±(708.7)	5.5	15.4 ±(4.2)	48.9	36.0 ±(4.9)
1UTG-M3	−48545.9	−46925.3 ±(1727.1)	6.4	13.6 ±(3.6)	51.1	39.7 ±(5.3)
1UTG-M4	−**63760.0**	−**62459.7** ±(787.0)	3.8	8.4 ±(2.6)	**63.2**	46.2 ±(7.6)
1UTG-M5	−45533.3	−44423.1 ±(689.1)	**3.3**	**7.2** ±(2.2)	**63.2**	**46.8** ±(8.4)
1UTG-R.	−122.4	−103.5 ±(6.0)	3.4	8.6 ±(3.2)	61.4	46.3 ±(8.8)
1WQC-M1	−13042.7	−12372.5 ±(751.2)	3.1	5.5 ±(1.3)	61.5	51.9 ±(5.5)
1WQC-M2	−13220.6	−12901.8 ±(235.6)	3.4	4.7 ±(0.6)	64.4	58.5 ±(3.1)
1WQC-M3	−13087.9	−12752.9 ±(222.7)	2.7	4.7 ±(0.9)	**70.2**	59.9 ±(4.3)
1WQC-M4	−**21553.8**	−**21434.3** ±(60.6)	2.7	4.1 ±(0.5)	69.2	**61.5** ±(2.9)
1WQC-M5	−13287.4	−13026.4 ±(126.9)	**2.5**	**4.0** ±(0.7)	69.2	61.1 ±(4.2)
1WQC-R.	−37.6	−26.9 ±(7.3)	*1.7	*2.3 ±(0.3)	*76.9	*71.1 ±(2.8)
1ZDD-M1	−21749.3	−20319.7 ±(774.8)	3.6	8.8 ±(2.2)	46.3	39.6 ±(3.1)
1ZDD-M2	−22342.8	−20152.4 ±(549.8)	2.7	5.5 ±(1.9)	47.8	43.0 ±(2.4)
1ZDD-M3	−20628.1	−19996.8 ±(286.6)	3.2	6.5 ±(2.2)	47.8	43.4 ±(2.1)
1ZDD-M4	−**30421.0**	−**28975.4** ±(851.2)	2.4	6.8 ±(2.5)	**48.5**	**43.7** ±(2.4)
1ZDD-M5	−20869.8	−20473.0 ±(242.3)	**1.9**	**3.6** ±(1.4)	**48.5**	43.6 ±(2.1)
1ZDD-R.	−57.5	−48.7 ±(4.8)	*0.8	*1.6 ±(0.8)	44.1	42.7 ±(1.2)
2MR9-M1	−25308.2	−24211.4 ±(718.9)	6.9	10.7 ±(2.0)	41.5	36.0 ±(3.5)
2MR9-M2	−26388.3	−25671.9 ±(439.8)	3.6	8.0 ±(1.7)	61.4	45.5 ±(5.8)
2MR9-M3	−26234.3	−25274.7 ±(461.5)	4.5	8.2 ±(1.6)	62.5	45.1 ±(6.1)
2MR9-M4	−**40690.2**	−**40346.8** ±(117.5)	3.1	6.7 ±(1.3)	63.1	**50.5** ±(4.7)
2MR9-M5	−26254.0	−25605.1 ±(385.7)	**2.6**	**5.9** ±(1.4)	**66.5**	49.5 ±(6.0)
2MR9-R.	−78.6	− 70.2 ±(4.9)	*1.4	*2.2 ±(0.6)	*83.5	*73.8 ±(5.7)
2P5K-M1	−28190.8	−18843.8 ±(2839.4)	10.5	15.4 ±(2.5)	45.2	29.7 ±(4.2)
2P5K-M2	−39455.7	−26651.9 ±(5408.1)	5.4	10.7 ±(2.8)	42.1	33.2 ±(3.2)
2P5K-M3	−33307.6	−25496.6 ±(4592.0)	5.7	12.5 ±(4.0)	39.7	32.4 ±(2.7)
2P5K-M4	−**55792.7**	−**49652.3** ±(3425.4)	5.9	10.0 ±(2.8)	40.5	34.2 ±(3.2)

(*continued on next page*)

Table 3 (*continued*)

ID_PDB	Energy		RMSD		GDT_TS	
	Lowest	Avg. (std)	Lowest	Avg. (std)	Highest	Avg. (std)
2P5K-M5	−39031.7	−30241.6 ±(6757.2)	**4.3**	**9.6** ±(3.7)	**45.6**	**35.0** ±(4.4)
2P5K-R.	−119.5	−100.3 ±(20.3)	*1.5	*2.5 ±(1.0)	*54.0	*50.8 ±(1.9)
2P6J-M1	−26462.4	−25180.0 ±(940.4)	8.9	14.6 ±(2.3)	44.7	33.4 ±(3.7)
2P6J-M2	−28137.2	−27691.5 ±(278.6)	5.1	10.2 ±(2.9)	56.7	46.0 ±(5.7)
2P6J-M3	−28004.7	−27332.8 ±(707.7)	3.5	11.0 ±(2.6)	60.1	45.9 ±(6.5)
2P6J-M4	−**47639.4**	−**47112.3** ±(230.7)	2.8	**4.7** ±(1.8)	**68.8**	**55.4** ±(5.9)
2P6J-M5	−28896.7	−28208.5 ±(728.4)	**2.7**	7.5 ±(2.4)	64.4	49.0 ±(4.7)
2P6J-R.	−93.6	−71.1 ±(19.9)	*2.2	*3.4 ±(1.4)	*74.5	*62.9 ±(5.6)
2P81-M1	−22134.2	−19988.3 ±(954.2)	3.8	8.1 ±(2.9)	36.9	31.7 ±(2.5)
2P81-M2	−23055.2	−22398.0 ±(683.1)	**2.9**	6.4 ±(1.9)	38.1	34.5 ±(1.9)
2P81-M3	−22902.2	−22395.1 ±(456.1)	3.3	**6.2** ±(1.9)	**39.2**	34.6 ±(1.8)
2P81-M4	−**40184.5**	−**39941.7** ±(119.6)	5.2	**6.2** ±(0.7)	36.9	34.7 ±(1.2)
2P81-M5	−23703.3	−23296.0 ±(238.1)	3.8	6.5 ±(1.1)	37.5	**35.8** ±(1.4)
2P81-R.	−75.2	−63.6 ±(4.5)	5.6	6.9 ±(0.7)	36.9	34.0 ±(1.2)
2PMR-M1	−52858.4	−51454.1 ±(828.4)	9.8	20.3 ±(5.8)	35.5	30.4 ±(2.4)
2PMR-M2	−54835.0	−53927.4 ±(658.8)	5.8	15.3 ±(5.5)	43.8	36.2 ±(3.2)
2PMR-M3	−54776.8	−53721.7 ±(299.8)	4.3	15.5 ±(4.1)	43.4	35.7 ±(2.3)
2PMR-M4	−**67768.8**	−**67313.3** ±(193.1)	3.3	7.4 ±(3.0)	47.0	40.2 ±(3.3)
2PMR-M5	−55570.7	−54641.0 ±(1149.7)	**2.5**	**6.4** ±(2.6)	**51.0**	**41.6** ±(3.6)
2PMR-R.	−141.8	−121.5 ±(19.1)	*1.4	*3.7 ±(0.9)	48.7	41.0 ±(3.3)
3V1A-M1	−32749.7	−32248.5 ±(399.4)	7.5	13.9 ±(2.6)	47.4	41.8 ±(3.4)
3V1A-M2	−33420.1	−33291.3 ±(75.5)	5.2	10.8 ±(2.5)	52.6	47.6 ±(2.1)
3V1A-M3	−33636.5	−33262.7 ±(168.1)	6.2	10.6 ±(2.9)	53.6	48.6 ±(2.2)
3V1A-M4	−**44105.9**	−**43850.9** ±(138.1)	2.3	4.5 ±(1.5)	**66.1**	**53.0** ±(4.9)
3V1A-M5	−33180.1	−32740.0 ±(278.3)	**1.9**	**3.3** ±(1.4)	60.9	51.8 ±(3.2)
3V1A-R.	−86.7	−76.6 ±(7.0)	*0.7	*2.4 ±(1.9)	55.7	51.4 ±(4.6)

the ability of the method to maintain the population diversity over the algorithm execution. It is noteworthy that the agents which comprise solutions with RG values far away from the optimum RG will always store bad solutions regarding the RMSD, but the method was exactly designed to generate and maintain different conformations and, consequently, through the combination of these diversities by the agents' interactions provide better solutions. Analyzing the three scenarios presented in Fig. 3, one can observe that M4 reached a good state space exploration as delineated by the proposed multimodal strategy and also kept a feasible trade-off between convergence and diversity of the individuals.

Thereby, we note that the multimodal strategy adopted in this work is not the only application possibility. Nevertheless, it is possible to conclude that regarding the distinct structural metric (RG) adopted, the partitioning of the conformational search space and the efforts to discover and optimize a set of distinct structural solutions enabled the improvement of the results.

– Comparisons regarding energy values: According to the lowest and average energy results in Table 3, it is possible to notice that the M4 outperformed all of the other methods including M5 for all of the targets. With this, it is clear that the multimodal strategies can significantly improve the effectiveness of the method over a roughness in the energy landscape. M4 was able to better explore the search space and find different energy basins (distinct structural models), while using the maintenance of the diversity of solutions enabled the improvement of the optimization performance. We note that we did not include Rosetta in this comparison of energy values because Rosetta contemplates multiple optimization stages where different energy functions are employed (i.e. the Rosetta models are not obtained using only one energy function). The entire Rosetta optimization process is based on various evaluation functions, which prevents the comparison with the other methods.

– Comparisons between methods M4, M5 and Rosetta: According to the results on Table 3, we observe that the methods M4 and M5 presented similar results. M4 reached better average results of RMSD and GDT_TS in 6 cases while M5 reached better average results in 4. In the other 6 cases, M4 and M5 obtained equal results or while one achieved a better average result of RMSD, the other performed better in GDT_TS, and vice versa. However, M4 produced better results than M5 just in 4 targets considering the lowest values of RMSD and 8 cases regarding the highest values of GDT_TS, although in some cases the differences were minimal. We emphasize that despite the similarities between M4 and M5, they comprise some different key components, such as the LS technique. Fig. 4 illustrates the comparison between the 3-D topology of the structures predicted by the proposed methods (M1, M2, M3, and M4), M5 and Rosetta superimposed upon the experimentally determined ones (red structures). From the Table 3, we notice that Rosetta outperformed all of the methods regarding the lowest and average values of RMSD in 11 targets and highest and average values of GDT_TS in 9 cases. Although it is possible to observe through a visual inspection (Fig. 4) that the methods M4, M5, and Rosetta obtained topologies (overall fold) very similar to each other and more close to the experimental ones.

To evaluate the statistical significance of these results (Table 3), we performed the *Mann-Whitney U test*, a non-parameteric pairwise comparisons procedure. Using a significance of $\alpha < 0.05$, we find that when we compared M3 and M4, differences in the predictions were not statistically significant only in 2 targets (1ZDD and 2P81), considering both RMSD and GDT_TS values. When comparing M4 and M5, differences in the results were not significant in most of the cases. However, the results of Rosetta when compared to M4 were statistically significant in almost all cases, except for proteins 1ACW and 1UTG, considering both RMSD and GDT_TS. We note that this evaluation corroborates with the previously made analysis. Details of the *p*-values of the statistical test can be found in the supplementary material.

Therefore, we can state that the proposed multimodal approach, the *TT-MMMA*, designed as an incremental algorithm using the combination of promising evolutionary components to address the PSP as a multimodal problem, is a contribution to the prediction of protein structures and that should be further explored to improve the results. The proposed method is capable of performing fast and effective predictions of protein 3-D structures when no known template structures and fold libraries are available. We only use

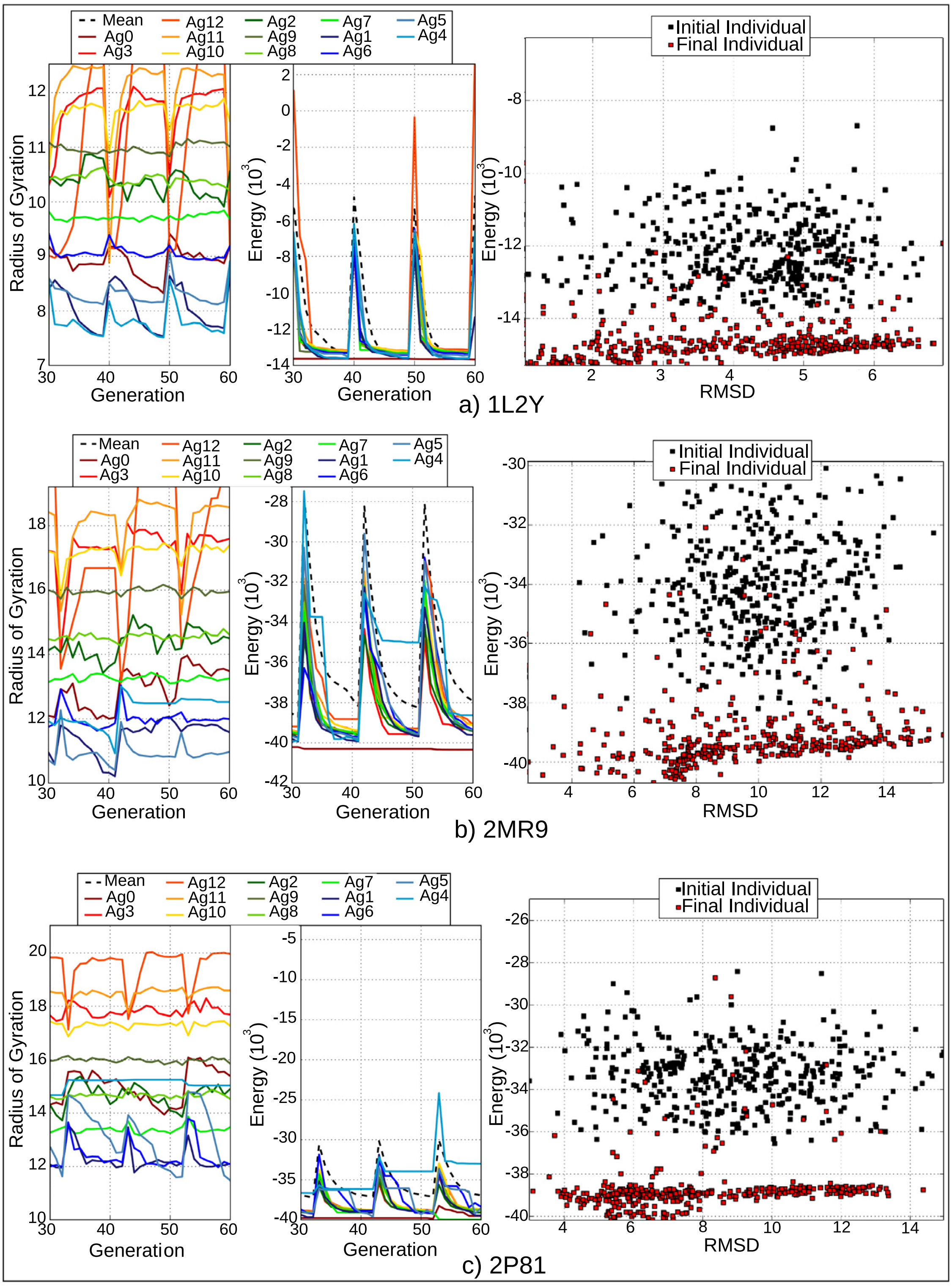


Fig. 3. Example of three scenarios of the optimization processes of the method M4 for three target proteins. (*i*) The leftmost plot shows the average RG of the sub-population of each agent throughout the algorithm execution. (*ii*) The central plot shows the energy convergence curve of the best solution of each subpopulation and the average energy of all of the agents' solutions. (*iii*) The rightmost plot shows a comparative analysis between the best solution of each subpopulation at the beginning of the optimization and the end of it, according to the energy and RMSD values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

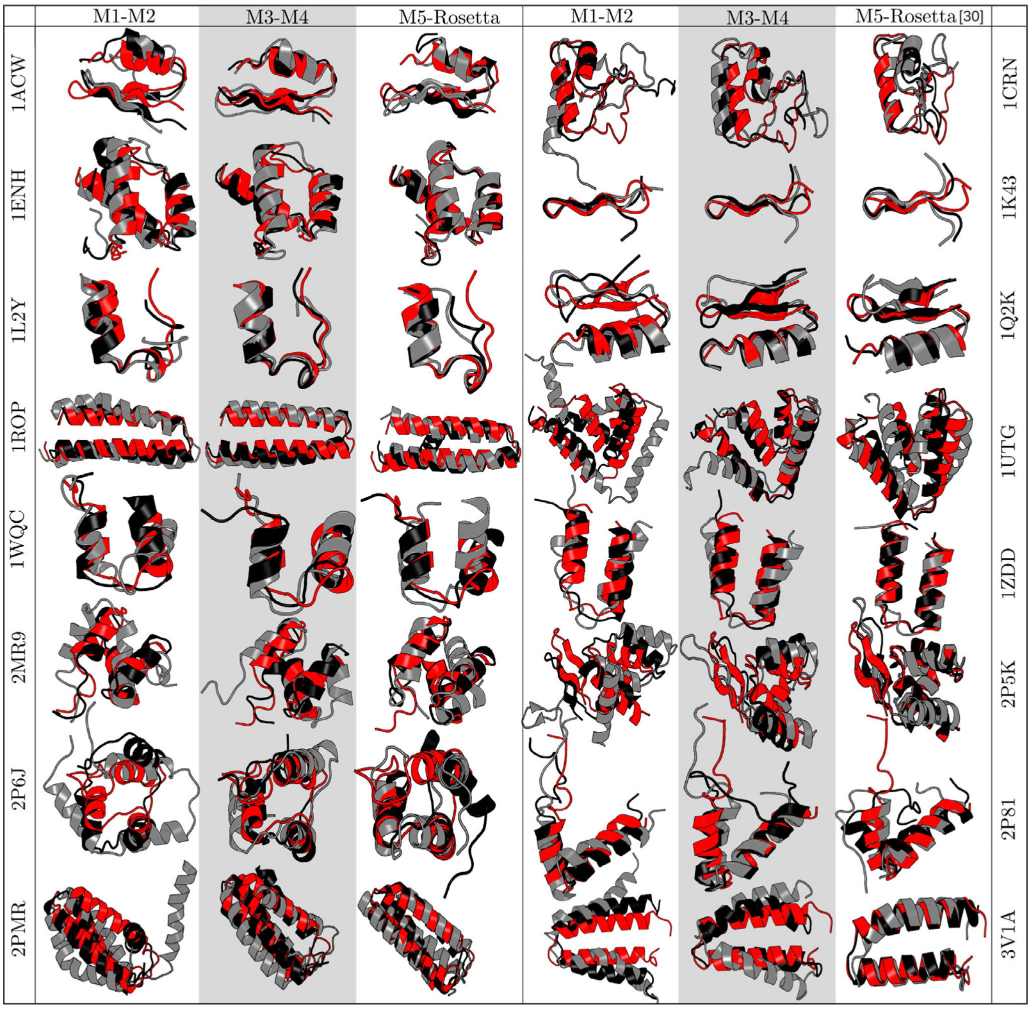


Fig. 4. Representation of the experimental structures (red) compared with the lowest RMSD predicted structures (black structures represent the methods at the left side of the legend and the gray ones represent the methods at the right side of the legend) for the *MA-SW-Chains* (M1), *Mod-MA* (M2), *TT-MA* (M3), *TT-MMMA* (M4), MA of Corrêa et al. (2016) (M5), and Rosetta (R.) algorithms. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the local information of conformational preferences of amino acid residues in proteins instead of fragments or segments of models obtained from experimentally-determined protein structures.

6. Conclusions

Despite the significant progress in the protein structure prediction field according to the latest CASP editions, it is still necessary to develop new strategies for extracting, representing and manipulating data from experimentally determined 3-D protein structures. It is also required to develop novel computational strategies to use this information to predict, only from the amino acid sequence of a protein, its corresponding 3-D structure. The development of computer prediction methods which reduce the computational effort and allow the prediction of the 3-D structure of proteins is presented as one of the main challenges in Structural Bioinformatics and Molecular Biology of the XXI century. There is an increasing need for new computational strategies that make use of previous knowledge and template information from experimentally determined protein structures to predict the unknown 3-D structure of proteins.

In this paper, we proposed three versions of a knowledge-based search strategy that rely on an incremental approach by using different components starting from a more general MACO, *MA-SW-Chains* algorithm, along with the ones described in the work by Corrêa et al. (2016) to deal with the PSP problem. The proposed versions (*Mod-MA, TT-MA* and *TT-MMMA*) use different population schemes and global search operators focused on the problem, allied to a local search technique to explore in a more effective way the protein conformational space. Since the PSP conformational space is known by its severe roughness and huge complexity due to the high dimensionality of variables, the last version of the proposed algorithms was developed to deal with the intrinsic multimodality of the problem by means of the exploration of multimodal optimization strategies.

As corroborated by experiments, the three algorithm versions outperformed the general described approach regarding biological significance quality through the RMSD and GDT_TS measures. The last version of the incremental approach was able to better guide the conformational space exploration and, consequently, find better solutions facing a multimodal and complex problem such as the PSP. The method overcomes the results of its previous versions, demonstrating the importance of adapting the method to deal with the multimodality issues of the problem by the generation and maintenance of the population diversity over the optimization process. Additionally, it can produce accurate predictions as the 3-D protein structures are conformationally comparable to their corresponding experimental ones. There are several research opportunities to be explored in this field, with relevant multidisciplinary applications in Computer Science and Bioinformatics. For instance, one could apply the proposed method to other classes of proteins. Likewise, other search techniques may be tested as variants of it. Finally, the experience gathered with known protein structures, knowledge-based operators and multimodal strategy can be improved to better tackle the problem.

Acknowledgments

This work was partially supported by grants from FAPERGS (002021-25.51/13, 16/2551-0000520-6), MCT/CNPq (473692/2013-9, 311022/2015-4), Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior (CAPES) (99999.000572/2016-00) - Brazil, Alexander von Humboldt-Stiftung (AvH) (99999.000572/2016-00) - Germany, and by Microsoft Corporation (CRM0740077) under a Microsoft Azure for Research Award.

References

Anfinsen, C.B., 1973. Principles that govern the folding of protein chains. Science 181 (4096), 223–230.

Baxevanis, A.D., Ouellette, B.F., 2004. Bioinformatics: APractical Guide to the Analysis of Genes and Proteins, 43, 2 John Wiley & Sons, Inc., New York, USA.

Belda, I., Madurga, S., Tarragó, T., Llorà, X., Giralt, E., 2007. Evolutionary computation and multimodal search: a good combination to tackle molecular diversity in the field of peptide design. Mol. Diversity 11 (1), 7–21.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I.N., Bourne, P.E., 2000. The protein data bank. Nucleic Acids Res. 28 (1), 235–242.

Borguesan, B., Inostroza-Ponta, M., Dorn, M., 2016. Nias-server: neighbors influence of amino acids and secondary structures in proteins. J. Comput. Biol. 24, 255–265.

Borguesan, B., e Silva, M.B., Grisci, B., Inostroza-Ponta, M., Dorn, M., 2015. APL: an angle probability list to improve knowledge-based metaheuristics for the three--dimensional protein structure prediction. Comput. Biol. Chem. 59, 142–157.

Boussaïd, I., Lepagnot, J., Siarry, P., 2013. A survey on optimization metaheuristics. Inf. Sci. 237, 82–117.

Bowie, J.U., Luthy, R., Eisenberg, D., 1991. A method to identify protein sequences that fold into a known three-dimensional structure. Science 253 (5016), 164–170.

Bradley, P., Misura, K.M., Baker, D., 2005. Toward high-resolution de novo structure prediction for small proteins. Science 309 (5742), 1868–1871.

Bryngelson, J.D., Onuchic, J.N., Socci, N.D., Wolynes, P.G., 1995. Funnels, pathways, and the energy landscape of protein folding: a synthesis. Proteins Struct. Funct. Bioinf. 21 (3), 167–195.

Chaudhury, S., Lyskov, S., Gray, J., 2010. Pyrosetta: a script-based interface for implementing molecular modeling algorithms using rosetta. Bioinformatics 26 (5), 689–691.

Chivian, D., Robertson, T., Bonneau, R., Baker, D., 2003. Ab initio methods. In: Structural Bioinformatics, 44. John Wiley & Sons, Inc, New Jersey, USA, pp. 547–557. chapter 27

Chou, K.-C., 1995. A novel approach to predicting protein structural classes in a (20–1)-d amino acid composition space. Proteins Struct. Funct. Bioinf. 21 (4), 319–344.

Consortium, .G.P., et al., 2015. A global reference for human genetic variation. Nature 526 (7571), 68–74.

Corrêa, L., Borguesan, B., Farfan, C., Inostroza-Ponta, M., Dorn, M., 2016. A memetic algorithm for 3-d protein structure prediction problem. IEEE/ACM Trans. Comput. Biol. Bioinf. PP (99). 1–1

Daggett, V., Fersht, A., 2003. The present view of the mechanism of protein folding. Nat. Rev. Mol. Cell. Biol. 4 (6), 497–502.

Das, S., Maity, S., Qu, B.-Y., Suganthan, P.N., 2011. Real-parameter evolutionary multimodal optimization-a survey of the state-of-the-art. Swarm Evol. Comput. 1 (2), 71–88.

Dawkins, R., 1976. The Selfish Gene, 1 Oxford university press, Oxford, UK.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6 (2), 182–197.

Desjarlais, J.R., Clarke, N.D., 1998. Computer search algorithms in protein modification and design. Curr. Opin. Struct. Biol. 8 (4), 471–475.

Dorn, M., Buriol, L.S., Lamb, L.C., 2011. A hybrid genetic algorithm for the 3-d protein structure prediction problem using a path-relinking strategy. In: 2011 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 2709–2716.

Dorn, M., Inostroza-Ponta, M., Buriol, L.S., Verli, H., 2013. A knowledge-based genetic algorithm to predict three-dimensional structures of polypeptides. In: 2013 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 1233–1240.

Dorn, M., e Silva, M.B., Buriol, L.S., Lamb, L.C., 2014. Three-dimensional protein structure prediction: methods and computational strategies. Comput. Biol. Chem. 53, 251–276.

Elofsson, A., Le Grand, S.M., Eisenberg, D., 1995. Local moves: an efficient algorithm for simulation of protein folding. Proteins Struct. Funct. Bioinf. 23 (1), 73–82.

Eshelman, L., 1993. Real-coded genetic algorithms and interval-schemata. In: Foundations of Genetic Algorithms, 2, pp. 187–202.

Garza-Fabre, M., Kandathil, S.M., Handl, J., Knowles, J., Lovell, S.C., 2016. Generating, maintaining and exploiting diversity in a memetic algorithm for protein structure prediction. Evol. Comput. 24 (4), 577–607.

Glibovets, N., Gulayeva, N., 2013. A review of niching genetic algorithms for multimodal function optimization. Cybern. Syst. Anal. 49 (6), 815–820.

Glover, F., 1994. Genetic algorithms and scatter search: unsuspected potentials. Stat. Comput. 4 (2), 131–140.

Guyeux, C., Côte, N.M.-L., Bahi, J.M., Bienia, W., 2014. Is protein folding problem really a np-complete one? first investigations. J. Bioinf. Comput. Biol. 12 (01), 1350017.

Handl, J., Lovell, S.C., Knowles, J., 2008. Investigations into the effect of multiobjectivization in protein structure prediction. In: International Conference on Parallel Problem Solving from Nature. Springer, pp. 702–711.

Heinig, M., Frishman, D., 2004. Stride: a web server for secondary structure assignment from known atomic coordinates of proteins. Nucleic Acids Res. 32 (suppl 2), W500–W502.

Hovmöller, S., Zhou, T., Ohlson, T., 2002. Conformations of amino acids in proteins. Acta Crystallogr. Sect. D-Biol. Crystallogr. 58 (5), 768–776.

Inostroza-Ponta, M., Farfán, C., Dorn, M., 2015. A memetic algorithm for protein structure prediction based on conformational preferences of aminoacid residues. In: Proceedings.... Genetic and Evolutionary Computation Conference (GECCO 2015). ACM, New York, USA, pp. 1403–1404.

Jayaram, B., Dhingra, P., Lakhani, B., Shekhar, S., 2012. Bhageerath-targeting the near impossible: pushing the frontiers of atomic models for protein tertiary structure prediction. J. Chem. Sci. 124 (1), 83–91.

Kabsch, W., Sander, C., 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22 (12), 2577–2637.

Kim, D.E., Blum, B., Bradley, P., Baker, D., 2009. Sampling bottlenecks in de novo protein structure prediction. J. Mol. Biol. 393 (1), 249–260.

Kim, D.E., Chivian, D., Baker, D., 2004. Protein structure prediction and analysis using the Robetta server. Nucleic Acids Res. 32, W526–531. Web Server issue

Kinch, L.N., Li, W., Monastyrskyy, B., Kryshtafovych, A., Grishin, N.V., 2016. Evaluation of free modeling targets in CASP11 and ROLL. Proteins Struct. Funct. Bioinf. 84 (S1), 51–66.

Kirkpatrick, S., Gelatt, C., Vecchi, M., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.

Kortemme, T., Morozov, A., Baker, D., 2003. An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein–protein complexes. J. Mol. Biol. 326 (4), 1239–1259.

Krasnogor, N., Smith, J., 2005. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. IEEE Trans. Evol. Comput. 9 (5), 474–488.

Kryshtafovych, A., Fidelis, K., Moult, J., 2014. CASP10 results compared to those of previous CASP experiments. Proteins Struct. Funct. Bioinf. 82 (S2), 164–174.

Kuhlman, B., Baker, D., 2000. Native protein sequences are close to optimal for their structures. Proc. Natl. Acad. Sci. U.S.A. 97 (19), 10383–10388.

Lazaridis, T., Karplus, M., 2000. Effective energy functions for protein structure prediction. Curr. Opin. Struct. Biol. 10 (2), 139–145.

Lobanov, M.Y., Bogatyreva, N., Galzitskaya, O., 2008. Radius of gyration as an indicator of protein structure compactness. J. Mol. Biol. 42 (4), 623–628.

Martí-Renom, M.A., Stuart, A.C., Fiser, A., Sánchez, R., Melo, F., Šali, A., 2000. Comparative protein structure modeling of genes and genomes. Annu. Rev. Biophys. Biomol. Struct. 29 (1), 291–325.

Molina, D., Lozano, M., García-Martínez, C., Herrera, F., 2010a. Memetic algorithms for continuous optimisation based on local search chains. Evol. Comput. 18 (1), 27–63.

Molina, D., Lozano, M., Herrera, F., 2010b. MA-SW-Chains: Memetic algorithm based on local search chains for large scale continuous global optimization. In: 2010 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 1–8.

Moscato, P., 1989. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report Caltech Concurrent Computation Program, Report. 826. California Institute of Technology, Pasadena, California, USA.

Moscato, P., Cotta, C., 2010. A modern introduction to memetic algorithms. In: Handbook of Metaheuristics, 146. Springer, Boston, MA, USA, pp. 141–183.

Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., Tramontano, A., 2016. Critical assessment of methods of protein structure prediction: progress and new directions in round XI. Proteins Struct. Funct. Bioinf. 84 (S1), 4–14.

Mühlenbein, H., Schlierkamp-Voosen, D., 1993. Predictive models for the breeder genetic algorithm i. Continuous parameter optimization. Evol. Comput. 1 (1), 25–49.

Neumaier, A., 1997. Molecular modeling of proteins and mathematical prediction of protein structure. SIAM Rev. 39 (3), 407–460.

Ong, Y.-S., Lim, M.H., Chen, X., 2010. Research frontier-memetic computation–past, present & future. IEEE Comput. Intell. Mag. 5 (2), 24.

Osguthorpe, D.J., 2000. Ab initio protein folding. Curr. Opin. Struct. Biol. 10 (2), 146–152.

Pruitt, K.D., Tatusova, T., Maglott, D.R., 2007. NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Res. 35 (suppl 1), D61–D65.

Richmond, T.J., 1984. Solvent accessible surface area and excluded volume in proteins: analytical equations for overlapping spheres and implications for the hydrophobic effect. J. Mol. Biol. 178 (1), 63–89.

Rocha, G.K., Custódio, F.L., Barbosa, H.J., Dardenne, L.E., 2016. Using crowding-distance in a multiobjective genetic algorithm for protein structure prediction. In: Proceedings.... Genetic and Evolutionary Computation Conference (GECCO 2016). ACM, New York, USA, pp. 1285–1292.

Rohl, C.A., Strauss, C.E., Misura, K.M., Baker, D., 2004. Protein structure prediction using Rosetta. Methods Enzymol. 383, 66–93.

Saleh, S., Olson, B., Shehu, A., 2013. A population-based evolutionary search approach to the multiple minima problem in de novo protein structure prediction. BMC Struct. Biol. 13 (Suppl 1), S4.

Shehu, A., Kavraki, L.E., Clementi, C., 2009. Multiscale characterization of protein conformational ensembles. Proteins Struct. Funct. Bioinf. 76 (4), 837–851.

Simons, K.T., Kooperberg, C., Huang, E., Baker, D., 1997. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. J. Mol. Biol. 268 (1), 209–225.

Solis, F., Wets, R.-B., 1981. Minimization by random search techniques. Math. Oper. Res. 6 (1), 19–30.

Song, Y., DiMaio, F., Wang, R.Y.-R., Kim, D., Miles, C., Brunette, T., Thompson, J., Baker, D., 2013. High-resolution comparative modeling with rosettacm. Structure 21 (10), 1735–1742.

Srinivasan, R., Rose, G.D., 1995. Linus: a hierarchic procedure to predict the fold of a protein. Proteins Struct. Funct. Bioinf. 22 (2), 81–99.

Syswerda, G., 1989. Uniform Crossover in Genetic Algorithms. In: Proceedings of the 3rd International Conference on Genetic Algorithms. International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, Inc., San Mateo, California, pp. 2–9.

Tai, C.-H., Bai, H., Taylor, T.J., Lee, B., 2014. Assessment of template-free modeling in CASP10 and ROLL. Proteins Struct. Funct. Bioinf. 82 (S2), 57–83.

Talbi, E.-G., 2009. Common concepts for metaheuristics. In: Metaheuristics: from Design to Implementation, 74. John Wiley & Sons, Inc., pp. 1–86. chapter 1

Tang, K., Li, X., Suganthan, P.N., Yang, Z., Weise, T., 2009. Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization. Technical Report. Nature Inspired Computation and Applications Laboratory, USTC, China.

Thomsen, R., 2004. Multimodal optimization using crowding-based differential evolution. In: 2004 IEEE Congress on Evolutionary Computation (CEC), 2, pp. 1382–1389.

Xu, D., Zhang, Y., 2012. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. Proteins Struct. Funct. Bioinf. 80 (7), 1715–1735.

Zemla, A., 2003. Lga: a method for finding 3d similarities in protein structures. Nucleic Acids Res. 31 (13), 3370–3374.

Zhang, W., Yang, J., He, B., Walker, S.E., Zhang, H., Govindarajoo, B., Virtanen, J., Xue, Z., Shen, H.-B., Zhang, Y., 2016. Integration of quark and i-tasser for ab initio protein structure prediction in casp11. Proteins Struct. Funct. Bioinf. 84 (S1), 76–86.

Zhang, Y., Skolnick, J., 2004. Scoring function for automated assessment of protein structure template quality. Proteins Struct. Funct. Bioinf. 57 (4), 702–710.