A dynamic evolutionary multi-agent system to predict the 3D structure of proteins

Leonardo Corrêa, Luciana Arantes, Pierre Sens, Mario Inostroza-Ponta, Márcio Dorn

To cite this version:

Leonardo Corrêa, Luciana Arantes, Pierre Sens, Mario Inostroza-Ponta, Márcio Dorn. A dynamic evolutionary multi-agent system to predict the 3D structure of proteins. WCCI 2020 - IEEE World Congress on Evolutionary Computation - CEC Sessions, Jul 2020, Glasgow / Virtual, United Kingdom. pp.1-8, 10.1109/CEC48606.2020.9185761 . hal-03132137

HAL Id: hal-03132137

https://hal.inria.fr/hal-03132137

Submitted on 4 Feb 2021

A dynamic evolutionary multi-agent system to predict the 3D structure of proteins

Leonardo Corrêa
Institute of Informatics
UFRGS
Porto Alegre, Brazil
llcorrea@inf.ufrgs.br

Luciana Arantes
DELYS, LIP6, INRIA
Sorbonne University
Paris, France
luciana.arantes@lip6.fr

Pierre Sens
DELYS, LIP6, INRIA
Sorbonne University
Paris, France
pierre.sens@lip6.fr

Mario Inostroza-Ponta
DIINF
USACH
Santiago, Chile
mario.inostroza@usach.cl

Márcio Dorn
Institute of Informatics
UFRGS
Porto Alegre, Brazil
mdorn@inf.ufrgs.br

***Abstract*—The protein structure prediction is one of the key problems in Structural Bioinformatics. The protein function is directly related to its conformation and the folding can provide to researchers better understandings about the protein roles in the cell. Several computational methods have been proposed over the last decades to tackle the problem. In this paper, we propose an *ab initio* algorithm with database information for the protein structure prediction problem. We do so by designing some versions of a multi-agent system that use concepts of dynamic distributed evolutionary algorithms to speed up and improve the optimization by better adapting the algorithm to the target protein. The dynamic strategy consists of auto-adapting the number of optimization agents according to the needs and current status of the optimization process. The system is able to scale in/out itself depending on some diversity criteria. The algorithms also take advantage of structural knowledge from the Protein Data Bank to better guide the search and constraint the state space. To validate our computational strategies, we tested them on a set of eight protein sequences. The obtained results were topologically compatible with the experimental correspondent ones, thus corroborating the promising performance of the strategies.**

***Index Terms*—optimization, multi-agent system, knowledge-based algorithm, structural bioinformatics**

I. Introduction

The research area concerned with the three-dimensional (3-D) protein structure prediction (PSP) configures a key issue in Structural Bioinformatics [1]. A single sequence of chained amino acids defines a protein that under specific physiological conditions folds into a particular conformation [2]. Proteins are in all living systems and perform an extensive set of fundamental life functions. The protein function's nature is directly related to its 3-D structure. Then, the protein folding provides to researchers better understandings about the protein roles in the cell [3]. The structural information corresponding to proteins can be obtained through experimental methods, such as X-ray crystallography and Nuclear Magnetic Resonance (NMR). However, such methods present some disadvantages, i.e., they are incredibly time-consuming and expensive. It is well known that the PSP, focused on the modeling just from the amino acid (aa) sequence, remains a challenge in the area. Some of the reasons in which this problem still imposes significant obstacles to scientists are due to the high cost and considerable required time of the experimental methods, high computational complexity and also by the lack of complete comprehension of the rules that conduct the biochemical processes and their relations over the protein folding [2], [3]. The problem is classified according to the computational complexity theory as an NP-hard problem due to the high dimensionality of variables and search space complexity [4]. The challenge relies on the combinatorial explosion of plausible conformations, where an aa chain can give rise to a few structures around native states among several possibilities.

The structure modeling as computational optimization can be seen as a way to overcome some of the PSP complexities and ease the protein structure-based studies. Therefore, several methods have been proposed to address the problem [1]. These methods can be classified, but not strictly, into two different classes, where they are grouped concerning the use or not of structural information from the Protein Data Bank (PDB) [5]: (*i*) first principle methods or *ab initio*; and (*ii*) fold recognition and comparative modeling methods. Specifically, in this work, we are interested in a group of methods located between these two classes, which consists in a hybrid class of knowledge-based methods that make use of template information from experimental protein structures associated with an *ab initio* strategy based on simulations of physicochemical properties of the folding process in nature [6]. Thus, to predict the 3-D structure of a protein adopting these concepts, a wide range of optimization metaheuristics are being proposed to find approximated solutions to the PSP [1]. Such techniques do not always guarantee the optimal solution, but they provide a reasonable approximation with a limited computational effort [7]. Also, the knowledge incorporation of protein structures from the PDB represents a critical strategy to support the modeling methods, reducing the conformational space size [8], [9].

Regardless of the computational prediction advances to deal with the problem, the PSP lasts a challenge in the area. The development of novel strategies and the incorporation of knowledge from experimentally determined protein structures combined with state-of-the-art methods is a real necessity [1]. Furthermore, especially in real problems as the PSP, the use of canonical metaheuristics does not always present the expected behavior. Some of the reasons are the severe roughness (multimodality) of the problem energy landscape, where even a small chain of amino acids can assume several conformations, and the difficulty in computationally represent the problem [10].

In this paper, we propose a distributed metaheuristic based on the concepts of autonomous agents and multi-agent systems (MAS) [11]. According to Merelli et al. [12], these concepts can be adopted as a tool to investigate the properties of biological systems that are difficult to study in more traditional ways, for example with in vitro experiments. It aims to efficiently explore the protein conformational space in a reasonable time, identifying native-like protein structures. We structured the presented method based on a previously proposed MAS for the PSP problem [13]. It has incorporated concepts of evolutionary algorithms (EAs) and the knowledge of known protein structures through the Angle Probabilist List (APL) strategy [14]. Then, we designed a MAS that implements a structured ternary tree population of agents as well as problem-specif components to deal with the problem, such as the use of contact maps information and the dynamic behavior based on the packaging of protein models. Each tree node (agent) of the system represents a computational process that has a subset of the population solutions. The agents exchange optimization information through global search operators. These connections tend to lead to the evolution and progressive improvements of the entire population. In EAs, ideas are represented by information exchanged between agents, which means the search operator's results. As in evolutionary culture, good ideas tend to survive while weak ones will disappear over the generations, culminating in a final set of reasonable solutions [15]. Thus, based on a previously proposed MAS of Corrêa et al. [13], we designed some MAS variations that explore auto-adapting concepts to distribute the system dynamically. It aims to rearrange the number of agents throughout the optimization, enabling or disabling processes based on predefined criteria regarding the problem optimization. Theoretically, the dynamic scaling in/out of the system can provide a better adaption of the metaheuristic (population) to the hyperparameters that control the prediction process since this dynamic concept of splitting in/out the agents was designated to auto-adapt the sub-populations regarding convergence/diversity criteria and prosperity of a given portion of the state space. However, as more control strategies implemented, more overhead the method presents. Thus, the paper also aims to analyze the advantages and drawbacks imposed by the algorithm's dynamic scaling in/out. The method also takes advantage of structural knowledge from the PDB, by using the APL and contact maps information in an attempt to constraint the conformational space and to better guide the EA [9]. It is noteworthy that the implemented distributed system can also be seen as a prototype for different metaheuristics. In this case, it is just necessary to adjust the search operators and the communication policies between agents to fit the heuristic needs.

Finally, our most significant contribution in this work is the design and assessment of different MAS versions, capable of adapt the metaheuristic to the targets, to deal with the PSP problem.

II. Problem Definition

The method versions presented in this paper use the same problem representation, energy function as fitness function, and the Angle Probability List strategy as shown in this section. The algorithms receive as input parameters the target protein aa sequence and its expected secondary structure (SS).

A. Protein Structure Representation

One of the existing possibilities to computationally represent the protein structure is by using its set of dihedral angles. It is based on the fact that bond lengths are nearly constant in an aa chain [16]. A peptide is a molecule composed of two or more amino acids chained by a chemical bond (peptide bond). Larger peptides are known as polypeptides or proteins. All amino acids found in nature present the same main structure (main chain or backbone) and differ in the side chain structure. Regarding the aa main chain, the peptide bond ($\texttt{C}-\texttt{N}$) (Omega angle - ω) has a partially-double bond feature and tends to be planar, presenting little or no modification. The free rotation is allowed around the bonds $\texttt{N}-\texttt{C}_\alpha$ (Phi angle - ϕ) and $\texttt{C}_\alpha-\texttt{C}$ (Psi angle - ψ), varying from -180° to +180° under a continuous domain. It is well known that this angles' set is the main responsible for the protein folding, whereas the stable local arrangements of amino acids generate its SS. As the backbone, the protein side chains also present dihedral angles, called Chi angles (χ). However, in this work, we adopted the centroid protein representation [6]. In such representation, the main chain remains fully atomic, but the representation of each aa side chain is simplified to a single pseudo-atom arranged in the side chain center of mass. The higher the number of features, the higher is the capacity of representing the protein as it appears in nature. Nevertheless, using all-atom models to represent proteins is computationally expensive, and thus, simplified representations are often used [17]. The centroid representation simplifies the side chain complexity, whereas keeping the overall protein folding by preserving the backbone integrity.

Therefore, the structure of a protein P with n amino acids is computationally represented by assigning the main chain dihedral angles to the amino acids that encompass the protein since the bond lengths between the atoms are not variable (1). We note that in this work, the computational representation used in the optimization processes is based on the backbone dihedral angles. But the model evaluations are performed using the Cartesian representation. We adopted the centroid objective function of Rosetta and the model conversion between the dihedral angle representation to the atomic coordinate is done by the own Rosetta's energy function implementation [6], [18].

$$P = (aa_1, ..., aa_{n-1}, aa_n) \tag{1}$$

$$aa_i = (\phi_i, \psi_i, \omega_i) \tag{2}$$

B. Objective Function

As objective function to evaluate the quality of a predicted protein model, we adopted the Rosetta energy function (centroid and minimization function) [6] provided by the PyRosetta toolkit [18]. The centroid Rosetta function considers more than ten weighted energy terms, most of them derived from knowledge-based potentials [6]. The energy value of the Rosetta function ($E_{rosetta}$) is given by the sum of all weighted function terms. The terms' weights are defined based on the Score3 Rosetta energy function. Additionally to the Rosetta terms, the SS term (3) [19] was included in the final energy function to support the secondary structures formation. The SS term aims to reinforce the corrected structures and penalize the uncorrected ones. The procedure gives a positive reinforcement ($const = 1000$) to the function if the SS (zp_i) corresponding to the i-th amino acid (aa_i) is equal to the SS (zi_i) of the same aa informed as input to the method. Otherwise, it gives a negative reinforcement ($const = -1000$) to the sum, when the SS of the corresponding amino acids are not the same. All target amino acids are compared over the fitness calculation. The DSSP method[1] was used to assign the secondary structures.

$$SS_{term} = \sum_{aa \in P} V(aa_i, zp_i, zi_i) \tag{3}$$

$$V(aa, zp, zi) = \begin{cases} \textit{-const}, & zp = zi \\ \textit{+const}, & zp \neq zi \end{cases} \tag{4}$$

In this work, besides the terms of the fitness function already described, we used a scheme to employ the information of contact maps (CMs) in the problem as a term of the energy function. The CM information is based on the knowledge discovery from experimental protein structure data. It tries to determine probabilistically which amino acids are in contact. In the last years, CMs have been used as a powerful addition to the PSP methods [9], [20]. As reported, improved contact methods can lead to enhanced protein structure predictors [8]. The CM term used in this work is based on an atom distance constraint function, which was previously proposed by Corrêa et al. [21]. It follows the same idea of weighting used in the SS term. In a CM, two amino acids are in contact, if the distance between their Cβ side chain atoms, or Cα of backbone for Glycine, is less than or equal to a distance threshold. A term of distance constraint is usually used to get the information from CMs and to overcome some inaccuracies of the energy function [22]. The CM term is a distance function between the amino acids in the CMs, and it aims to positively reinforce the aa pairs that are within the contact bounds or to penalize the ones that are out of the threshold, according to (5). So, we employed a reduced list of $L/2$ medium and long range predicted contacts. The CMs were predicted by the MetaPSICOV predictor [23].

$$CM = \sum_{i,j}^{CM_{L/2}} = \begin{cases} p \times -c, & d(i,j) \leq ub \\ p \times -c \div 2, & ub < d(i,j) \leq ub + 2 \\ p \times +c, & d(i,j) > ub + 2 \end{cases} \tag{5}$$

where p denotes the probability of amino acids are in contact, c is a constant, ub is a residue contact upper bound, and $d(i,j)$ represents the Euclidean distance between a pair of amino acids in the predicted contact list. Following the literature, we adopted the contact threshold of $ub = 8$Å. For the constant c, we adopted $c = 1000$ to follow the reinforcement values defined in the SS term (3).

Finally, all the terms described were integrated to the Rosetta function composing the evaluation function (E_{final}) (6) used in this work.

$$E_{final} = E_{rosetta} + SS_{term} + CM_{term} \tag{6}$$

C. Conformational Preferences of Amino Acids

The developed MAS considers the experimental knowledge stored in the PDB by the Angle Probability List strategy. The main reason for incorporating such information to the method is to constraint the conformational search space. The APL[2], proposed by Borguesan et al. [24] and extended by Corrêa et al. [19], aims to assign the angle values to the target amino acids through analysis of the conformational preferences of these amino acids in experimentally determined structures according to their secondary structures and arrangements. To adopt the structural information of known protein templates, concerning the authors, they built histograms of $[-180°, 180°] \times [-180°, 180°]$ cells for each aa and SS, generating combinations up to 9 amino acids (1-9) and their secondary structures, and considering the reference aa neighborhood for combinations larger than 1. We note that the angle values are attributed only to the reference aa. Each histogram cell (i,j) has the number of times that a given aa (or combination of amino acids) presents a torsion angles pair ($i \leq \phi < i+1$, $j \leq \psi < j+1$) concerning a SS. The APL was calculated for each histogram, representing the normalized frequency of each cell. APL was incorporated in the methods to create short combinations of amino acids aiming the high-quality individual initialization as a starting point for the optimization and after a restarting function. A weighted random selection was employed to select the angle values from APL. It gives higher chances to the histograms' cells that present a higher relative frequency of occurrence. For a full APL description, we refer to the web server NIAS-Server[3] [14] created to investigate the amino acids conformational preferences.

III. Proposed Algorithm

To deal with the PSP problem, we propose some variations of a MAS that incorporate the experimental knowledge from the PDB's protein structures, such as the APL strategy and contact maps information, and concepts of population-based EAs [11], [25]. The algorithm was structured based on a previously proposed MAS for the problem [13]. MAS are used to tackle complex problems in a distributed way, devising tasks among agents, and exploring a decentralized approach. An agent is an independent computational process that, under some circumstances, interacts with other agents to solve a given task [11]. The multi-agent paradigm has been shown a useful approach for problems that present repetitive and time-consuming tasks, knowledge share and management, such as modeling of complex systems [12]. These concepts make the

[1] https://swift.cmbi.umcn.nl/gv/dssp/

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

[3] http://sbcb.inf.ufrgs.br/npas

paradigm suitable to simulate biological systems that can be decomposed in several independent but interacting entities, each one represented by an agent [26].

Besides that, we designed a distributed dynamic strategy of auto-adapting and rearranging the number of agents in the system throughout the optimization. It aims to create or disable processes (scaling in/out of the system) based on convergence/diversity criteria and solution improvements. We believe that this dynamic idea of splitting in/out the system agents based on criteria related to the optimization problem, such as quality of solutions or population convergence, can provide a better adaption of the metaheuristic to the hyperparameters that control the process according to the target protein characteristics. Also, the dynamic behavior can save computational resources when they are not need during the optimization, disabling processes, and just creating when necessary. On the contrary, static systems with a predefined number of agents will always consume all the available resources even when it is not really necessary, e.g., when modeling an easier target protein or a convergence was reached. Hence, the auto-adapting concept was implemented to allow the scaling of the agents, leaving to the system itself the responsibility for control this dynamic behavior. Based on such approach, we developed some MAS versions by exploring a distinct number of agents and incorporating the dynamic concept into the system to analyze their performance, scalability, and impact on the protein prediction results together with the designed problem-specific strategies.

As shown in Figure 1, our method has a set of agents structured in a tree-based data structure, where each one performs specific tasks and interacts in a cooperative coevolution scheme to reach reasonable solutions to the problem. The algorithm uses the APL strategy, already explained in a previous section, on the initialization of solutions to reduce the search space by incorporating high-quality solutions as a starting point to the metaheuristic. The MAS steps are described in the sections below.

A. Implementation

The proposed MAS was implemented in Python over the Multiprocessing native library, which is a process-based threading interface[4]. The communication between agents is done through the shared memory approach. All simulations of the MAS were performed in the same cluster. We did not consider process faults.

B. Individual Representation

Each individual of the metaheuristic's population represents a possible solution to the problem. Each amino acid of the target protein means a set of three values: ϕ, ψ and ω dihedral angles since we are using the centroid structure representation (Sec. II-A). With this, a solution for a target with n amino acids is computationally encoded by a vector of real values of size $n \times 3$, following (1). The population's encoding scheme is represented by a vector with the metaheuristic's individuals.

C. Proposed Metaheuristic and Optimization

In this work, the APL strategy was used to initialize the solutions of the metaheuristic, by generating different amino acid combinations (length of 1-3 aa and 5-9 aa), in an attempt to feed the algorithm with high-quality solutions when compared to those randomly initialized (Sec. II-B).

According to the Figure 1, the main process (Fig. 1-A) is responsible for starting the optimization agents and finish the system execution. In the optimization step, each optimization agent (OA) can be seen as a part (sub-population) (Fig. 1-B) of the global evolutionary metaheuristic. We incorporated, as behaviors of the optimization agents, concepts of evolutionary-based algorithms, such as crossover and swap operators (Fig. 1-D). The interactions among the agents through search operators lead to population evolution and progressive improvements. All interactions are performed by shared memory and critical section instructions. Each agent has its cycles or generations. We note that all the crossover operations are done by the Secondary Structure Uniform crossover, explained in the work of Corrêa et al. [19].

Initially, we employed a population of thirteen OA organized in a hierarchical ternary tree, which are the agents 0-12 illustrated in Figure 1. Each OA has a set of thirty solutions where one of them is called *current* solution, and the others are the *pockets* (Fig. 1-B). The population is organized in overlapped sub-populations composed of three *supporters* and one *leader* agent. An agent can only interact with the leader agent of the sub-population to which it belongs. The OA has some defined tasks to be done:

- At the initialization phase, all agents set their pocket solutions to the APL strategy (Fig. 1-B);
- In each generation, the agents do n_{cross}=10 inner crossover operations with the solutions in their respective pockets. The offspring are also stored in the current solution of the agents (Fig. 1-C);
- At every ten generations, the leader agent of a sub-population makes crossover with agents located in the lower level. The offspring is stored into the current solution of the lower level agent (Fig. 1-D);
- Each agent keeps the pockets always sorted;
- The agent updates the population in each generation. This task can be divided in small steps. First, the current solution is stored in one of the pockets if it is better than one that is already stored. Second, if the agent is in the lower level of a sub-population, then it sends the best solution to the leader agent (swap operation). Therefore, the best solutions are kept on the top of the hierarchy in the *Agent 0* pockets, diversifying the solutions from different sub-population patterns (Fig. 1-D);
- At the end of each generation, each agent discards the worst solution in the pockets ($CV < 10$, Sec. III-D) and creates a new one to avoid premature convergence and escape from local minimal.

D. The Dynamic Multi-agent System

The MAS presented in this work was structured based on a previously proposed method for the problem [13]. There-

[4] https://docs.python.org/2/library/multiprocessing.html

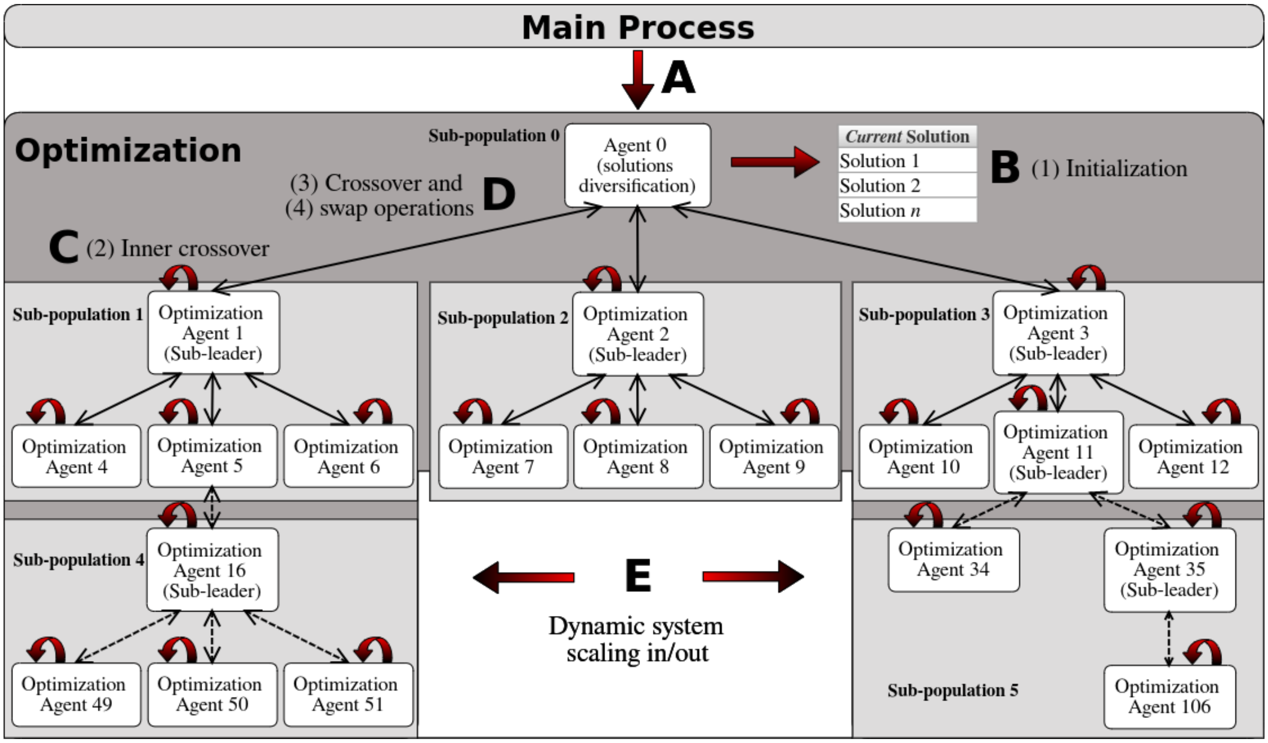


Fig. 1: Structure, actions and interactions of the dynamic MAS.

fore, we modified the previous static algorithm implementing auto-adapting concepts to distribute the system dynamically throughout the optimization. The dynamic strategy consists in to rearrange the number of optimization agents, enabling or disabling processes based on predefined optimization criteria. So, we call the system's ability to create or disable agents as scale in and scale out, as illustrated in Figure 1-E. In this work, the system scales based on two criteria: convergence of the population and solution improvements as denoting the prosperity of a given portion of the search space. The convergence criteria are related to the coefficient of variation (CV) [27] of the average radius of gyration (RG) of an ensemble of solutions. This ensemble of solutions means a sub-population of a given agent. The CV is a measure of relative variability to estimate the sample diversity degree. The lower CV indicates that the sample tends to be similar. The RG of a protein structure is defined as the quadratic mean distance between all the protein atoms and its center of mass and can be used as a packaging indicator, since the lower the RG, the greater the proximity of the atoms with the protein center of mass [28]. In this way, each agent is responsible for create new agents and disable itself based on some conditions, as follows:

- An agent scales out (i.e., creates another agent), if and only if,
  1) Its sub-population's CV is less than $p\%$ (generally p=10 [27]);
  2) It is able to create a new sub-agent. It creates a new agent, if and only if,
     a) It is a leaf in the hierarchical ternary tree (Fig. 1);
     b) It satisfies the maximum allowed number of children (MNC=3 - ternary tree) per node;
     c) It satisfies the minimum and maximum allowed number of total agents in the system (Table II).
- An agent is disabled (disable itself), if and only if,
  1) It is a leaf in the hierarchical ternary tree (Fig. 1);
  2) Its CV is less than 5%, denoting a severe convergence;
  3) Its best solution has not been improved during a number l=10 of generations.
- The agent only creates one child per time;
- The checking for scaling is done at the end of every agent generation.

IV. Computational Experiments

The algorithms presented in this work were run eight times with a stop criterion of 10^6 energy (fitness) evaluations per run on each target protein, regardless of the number of agents considered in the simulation. We used as case studies in our tests the amino acid sequences of 8 PDB's target proteins described in Table I. To analyze the performance of the proposed strategies, we developed some MAS versions with a different number of agents and implementations (static or dynamic). Table II summarizes the differences among them. We note that the methods M4 and M5 are the dynamic ones (Section III-D). They use 13 agents as minimum allowed number of total agents in the system, which is the base system. Throughout the optimization, they cannot surpass the maximum allowed number of total agents in the system, which is 25 agents for M4 and 40 for M5. However, the tree structure may become unbalanced over the process, with more agents in one brunch than in another. On the other hand, the static versions are unchangeable and keep the same data structure all over the optimization process. They follow the same structure shown in Figure 1, only increasing the number of agents in ascending order. To evaluate the developed algorithms regarding the most relevant methods in the field, we compared them with the Rosetta *ab initio* protocol (M6) [6], one of the most promising approaches to deal with the problem [8]. Obtained results are presented in the next section.

TABLE I: Target amino acid sequences.

Protein	Length	SS Content
`1AB1` (Fig.4a)	46	One β-sheet/Two α-helices
`1ACW` (Fig.4b)	29	One β-sheet/One α-helix
`1AIL` (Fig.4c)	70	Three α-helices
`1DFN` (Fig.4d)	30	One β-sheet
`2MR9` (Fig.4e)	44	Three α-helices
`2P5K` (Fig.4f)	64	One β-sheet/Three α-helices
`3V1A` (Fig.4g)	48	Two α-helices
`T0820-D1` (Fig.4h)	90	Three α-helices

TABLE II: Components used in the MAS variations.

Method	Implementation	Total number of agents
`M1`	Static	13 (base system)
`M2`	Static	25 (4 incomplete levels)
`M3`	Static	40 (4 levels)
`M4`	Dynamic	Min.=13, Max.=25
`M5`	Dynamic	Min.=13, Max.=40
`Rosetta`	-	-

A. *Results and Discussion*

For each method in Table II, we analyzed its execution time, scalability, and structural analysis of the obtained results regarding biological measures. Figure 2 illustrates the average running time, in seconds, regarding the eight executions for each target protein and MAS version. By analyzing the plots, it is possible to note that the designed MAS was able to reach reasonable scalability since the first version M1 was the slowest for all target proteins. It is shown as the number of agents increases, regarding the MAS variations, the running time decreases due to the higher number of agents running in parallel. Corroborating, the M3 and M5 were the fastest for all targets. We notice that the static versions, when comparing M2 (static) against M4 (dynamic) and M3 (static) against M5 (dynamic), were faster than the dynamic ones. We believe that this is due to the small overhead imposed by the dynamic scaling of the system, such as the checking procedure for the scaling performed by each agent. Figure 3 shows two different scenarios regarding the implemented dynamic strategy. The plots at the left show the total number of agents used throughout the optimization executions of the method M5. Each color represents an execution. We observe that the total number of agents varies all over the optimization, showing that different target proteins and runs present distinct behavior and, consequently, computational needs. Therefore, it is observable that the dynamic behavior can save computational resources when they are not need during the optimization, disabling processes and just creating when necessary. Plots at the right illustrate the dynamic scaling in/out of the agents over a single execution of the algorithm M5. The **x**-axis denotes the agent ID regarding its position in the tree structure. Analyzing them, we observe that for a single execution, there is a lot of variation in terms of creation and disabling of agents. This corroborates with the idea that the dynamic strategy can provide a better adaption of the metaheuristic to the hyperparameters that control the process according to the target protein characteristics.

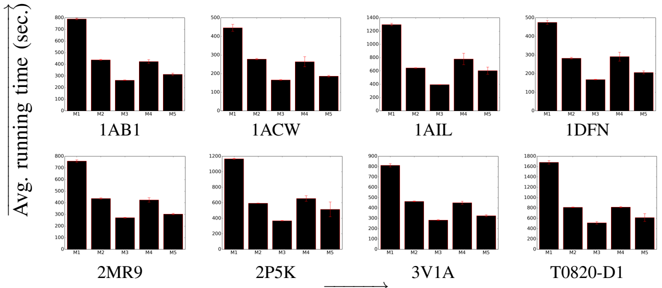


Fig. 2: Average running time (**y-axis**) regarding the eight algorithm executions for each target protein and MAS version (**x-axis**) (M1, M2, M3, M4 and M5, respectively).

For each target protein, we analyzed the best solutions among the performed runs regarding the root-mean-square deviation (RMSD, minimization measure) and the global distance total score test (GDT_TS, maximization measure) of the predicted structures in comparison with their corresponding experimental ones. Table III describes the achieved optimization results of the MAS versions and method of Rosetta applied to the case studies. According to the results summarized in the Table III, we observe that the dynamic and static versions did not present significant differences in the average of the cases. Both the static and dynamic versions achieved better average results of RMSD in 4 cases. However, the static versions reached better average results of GDT_TS in 5 cases. These results show that the dynamic strategy or the increased number of agents are not enough to significantly improve the results when analyzing the final structures from a biological point of view.

Also, Figure 4 shows the comparison between the 3-D topology of the models predicted by method M5 (blue) and Rosetta (gray) superimposed upon the experimentally determined structures (red). Observing the results in Table III, we note that Rosetta overcomes all of the other methods regarding the average RMSD values in 5 targets and regarding the average GDT_TS in 4 cases. Although it is observable by visual inspection of Figure 4 that the M5 and Rosetta reached overall target folding very similar to each other and comparable to the experimentally determined structures. Thus, we can state that the proposed dynamic MAS is a promising contribution to the prediction of protein structures and that should be further explored to improve the biological results.

TABLE III: Algorithms simulation results. The **boldface** numbers are the best results regarding RMSD and GDT_TS. The (*) denotes the best results between only the MAS variations.

Method	RMSD (Å)							
	Lowest	Avg. (std.)	Lowest	Avg. (std.)	Lowest	Avg. (std.)	Lowest	Avg. (std.)
	1AB1		1ACW		1AIL		1DFN	
M1	2.21	**3.56*** ± (0.93)	1.9	2.5* ± (0.48)	4.68	**5.72*** ± (0.99)	1.98	2.74 ± (0.67)
M2	1.87	3.64 ± (1.26)	2.52	3.46 ± (0.56)	4.58	8.35 ± (2.39)	2.14	3.3 ± (0.88)
M3	**1.75***	3.76 ± (1.21)	2.56	3.23 ± (0.65)	4.93	6.71 ± (1.33)	1.92	**2.56*** ± (0.38)
M4	2.38	4.27 ± (1.31)	1.85*	3.4 ± (0.88)	**4.51***	6.85 ± (2.1)	**1.75***	2.72 ± (0.79)
M5	2.75	4.36 ± (0.95)	2.09	3.0 ± (0.62)	4.64	7.17 ± (2.49)	2.38	4.27 ± (3.48)
Rosetta	3.45	5.55 ± (1.02)	**1.66**	**2.11** ± (0.38)	6.85	9.45 ± (1.05)	3.63	5.29 ± (0.86)
Method	**2MR9**		**2P5K**		**3V1A**		**T0820-D1**	
M1	1.79*	2.3 ± (0.36)	2.99	4.42 ± (1.31)	2.21*	2.9 ± (0.37)	9.19	11.96 ± (2.01)
M2	1.97	2.25 ± (0.27)	2.63	4.11 ± (0.99)	2.56	3.15 ± (0.34)	9.04	11.52 ± (1.89)
M3	2.07	2.46 ± (0.3)	2.7	4.63 ± (0.89)	2.69	3.09 ± (0.28)	9.42	11.87 ± (1.35)
M4	2.03	2.24* ± (0.21)	2.6*	3.7* ± (0.77)	2.41	2.84* ± (0.2)	9.05	10.96 ± (1.7)
M5	1.99	3.03 ± (1.05)	3.03	3.96 ± (1.21)	2.36	2.93 ± (0.33)	8.43*	10.6* ± (1.94)
Rosetta	**1.43**	**2.22** ± (0.69)	**1.57**	**2.29** ± (1.0)	**0.7**	**2.51** ± (1.9)	**7.34**	**9.19** ± (1.7)
Method	**GDT_TS**							
	Highest	**Avg. (std.)**	**Highest**	**Avg. (std.)**	**Highest**	**Avg. (std.)**	**Highest**	**Avg. (std.)**
	1AB1		**1ACW**		**1AIL**		**1DFN**	
M1	73.91	69.84 ± (3.53)	72.41	67.78* ± (3.76)	**57.14***	**50.45*** ± (4.96)	**59.17***	**52.09*** ± (2.98)
M2	78.26	**71.81*** ± (4.0)	72.41	61.42 ± (4.39)	52.5	41.74 ± (6.27)	52.5	48.96 ± (3.25)
M3	**79.35***	69.57 ± (5.31)	65.52	62.61 ± (3.28)	52.86	45.63 ± (4.84)	58.33	52.08 ± (3.03)
M4	72.83	66.31 ± (5.27)	73.28	60.99 ± (6.27)	56.07	47.54 ± (7.3)	55.83	50.41 ± (3.0)
M5	70.65	66.78 ± (2.28)	75.0*	64.76 ± (5.44)	53.21	48.08 ± (4.28)	53.33	48.12 ± (5.66)
Rosetta	62.5	56.45 ± (4.27)	**77.59**	**73.49** ± (3.33)	48.93	39.33 ± (5.36)	49.17	44.69 ± (2.6)
Method	**2MR9**		**2P5K**		**3V1A**		**T0820-D1**	
M1	78.98*	73.3 ± (2.88)	51.98*	47.57 ± (2.71)	50.0	49.03 ± (0.71)	42.4	34.17 ± (3.96)
M2	76.14	73.15 ± (1.6)	49.6	47.82 ± (1.42)	52.07	48.89 ± (1.57)	40.28	35.28* ± (4.16)
M3	76.14	73.08 ± (2.09)	49.6	46.68 ± (1.94)	51.04	48.83 ± (1.34)	38.33	33.4 ± (3.02)
M4	78.41	**74.22*** ± (3.03)	51.59	47.72* ± (2.51)	52.08*	50.45* ± (1.39)	40.28	34.3 ± (3.72)
M5	77.84	68.39 ± (7.95)	49.6	47.57 ± (1.9)	52.08*	49.28 ± (1.45)	42.5*	33.82 ± (4.32)
Rosetta	**83.52**	73.79 ± (6.59)	**53.97**	**51.54** ± (1.85)	**55.21**	**51.44** ± (4.63)	**45.28**	**39.62** ± (3.71)

V. Conclusion

It is well known that there is an increasing need for new computational strategies able to reach the best potential of prediction methods and the structural information from protein databases and its use in the protein structure prediction. In this paper, we proposed a dynamic multi-agent system that incorporated in the prediction process the structural knowledge from the PDB through the APL strategy and CMs information. It also explored concepts of population-based evolutionary algorithms for the PSP problem. We analyzed different MAS versions and processes distribution as well as the method dynamic proposal. As corroborated by experiments, results show that the proposed MAS has good scalability in terms of computational performance and promising ability to predict good approximations to the 3-D protein structures regarding the structural analysis.

Acknowledgment

This work was supported by grants from MCT/CNPq [*311611/2018-4*], FAPERGS, Alexander von Humboldt-Stiftung (AvH) [*BRA 1190826 HFST* CAPES-P] - Germany, and was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001. CAPES STICAMSUD - Brazil.

References

[1] M. Dorn, M. B. e Silva, L. S. Buriol, and L. C. Lamb, “Three-dimensional protein structure prediction: methods and computational strategies,” *Comput. Biol. Chem.*, vol. 53, pp. 251–276, 2014.

[2] C. B. Anfinsen, “Principles that govern the folding of protein chains,” *Science*, vol. 181, no. 4096, pp. 223–230, 1973.

[3] K. A. Dill and J. L. MacCallum, “The protein-folding problem, 50 years on,” *Science*, vol. 338, no. 6110, pp. 1042–1046, 2012.

[4] C. Guyeux, N. M.-L. Côte, J. M. Bahi, and Bienia, “Is protein folding problem really a np-complete one? first investigations,” *J. Bioinf. Comput. Biol.*, vol. 12, no. 01, p. 1350017, 2014.

[5] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, “The protein data bank,” *Nucleic Acids Res.*, vol. 28, no. 1, pp. 235–242, 2000.

[6] C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker, “Protein structure prediction using rosetta,” *Methods Enzymol.*, vol. 383, pp. 66–93, 2004.

[7] E.-G. Talbi, “Common concepts for metaheuristics,” in *Metaheuristics: from design to implementation*. John Wiley & Sons, Inc., 2009, vol. 74, ch. 1, pp. 1–86.

[8] L. A. Abriata, G. E. Tamò, B. Monastyrskyy, A. Kryshtafovych, and M. Dal Peraro, “Assessment of hard target modeling in casp12 reveals an emerging role of alignment-based contact prediction methods,” *Proteins: Struct. Funct. Bioinf.*, vol. 86, pp. 97–112, 2018.

[9] B. Kuhlman and P. Bradley, “Advances in protein structure prediction and design,” *Nat Rev Mol Cell Biol*, pp. 1–17, 2019.

[10] J. Handl, S. C. Lovell, and J. Knowles, “Investigations into the effect of multiobjectivization in protein structure prediction,” in *International Conference on Parallel Problem Solving from Nature*. Springer, 2008, pp. 702–711.

[11] N. R. Jennings, K. Sycara, and M. Wooldridge, “A roadmap of agent research and development,” *Auton agent multi-ag*, vol. 1, no. 1, pp. 7–38, 1998.

[12] E. Merelli, G. Armano, N. Cannata, F. Corradini, M. d’Inverno, A. Doms, P. Lord, A. Martin, L. Milanesi, S. Möller *et al.*, “Agents in bioinformatics, computational and systems biology,” *Brief Bioinform*, vol. 8, no. 1, pp. 45–59, 2006.

[13] L. de Lima Corrêa, M. Inostroza-Ponta, and M. Dorn, “An evolutionary multi-agent algorithm to explore the high degree of selectivity in three-dimensional protein structures,” in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 1111–1118.

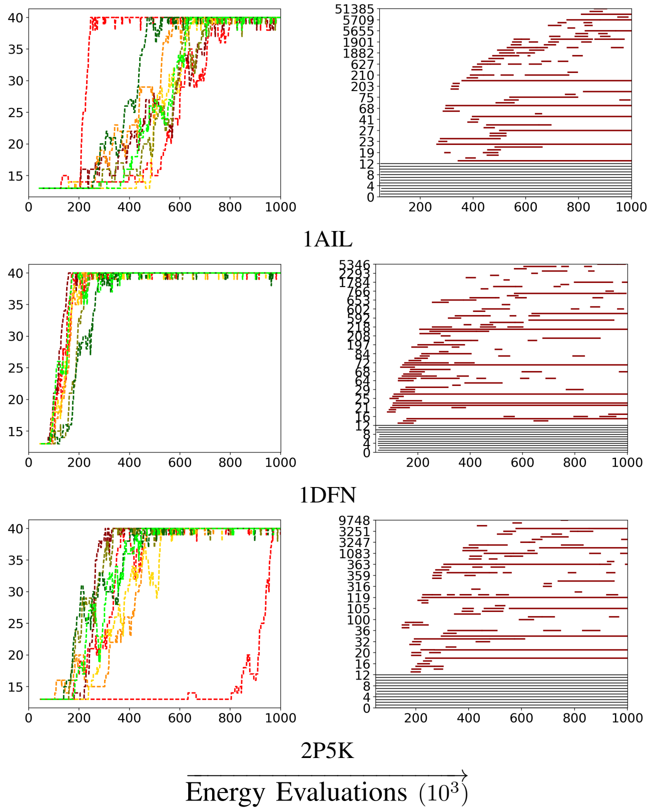
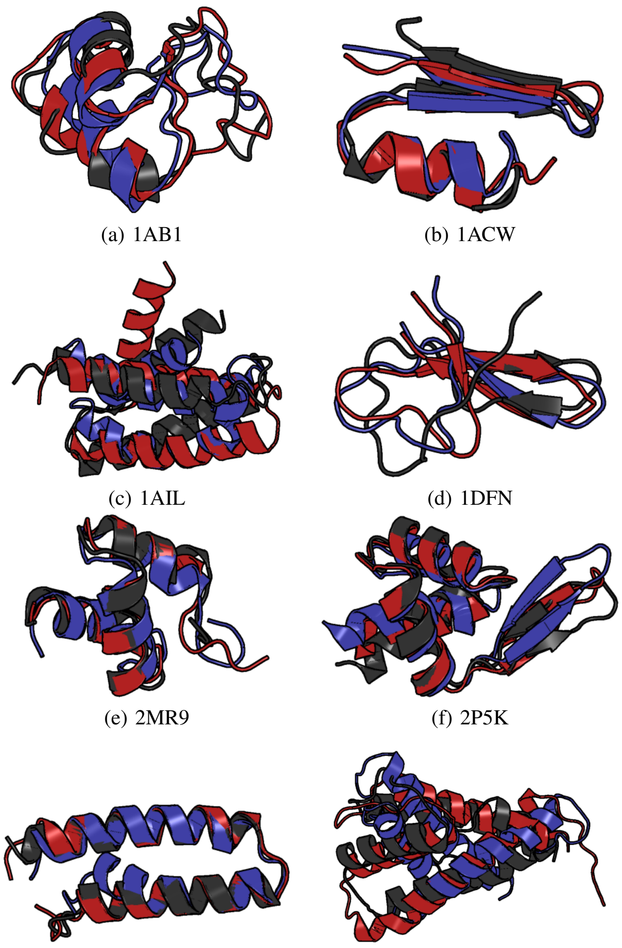


Fig. 3: Plots at the left show the total number of agents (**y-axis**) used throughout the eight runs of the method M5 for the illustrated targets, which can create up to 40 agents. Each color represents an execution. Plots at the right illustrate the dynamic scaling in/out of the agents over a single run of the M5. The **y-axis** for the plots at right denotes the agent ID regarding its position in the tree structure.



(a) 1AB1 (b) 1ACW (c) 1AIL (d) 1DFN (e) 2MR9 (f) 2P5K (g) 3V1A (h) T0820-D1

Fig. 4: Cartoon representation of the experimental structures (red) compared with the lowest RMSD of the predicted ones for the M5 (blue) and Rosetta (gray) algorithms. Graphic representation was prepared with PyMOL [29].

[14] B. Borguesan, M. Inostroza, and M. Dorn, "Nias-server: Neighbors influence of amino acids and secondary structures in proteins," *J. Comput. Biol.*, vol. 24, pp. 255–265, 2016.

[15] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Trans. Evol. Comput.*, vol. 9, no. 5, pp. 474–488, 2005.

[16] A. Neumaier, "Molecular modeling of proteins and mathematical prediction of protein structure," *SIAM review*, vol. 39, no. 3, pp. 407–460, 1997.

[17] D. Chivian, T. Robertson, R. Bonneau, and D. Baker, "Ab initio methods," in *Structural Bioinformatics*. New Jersey, USA: John Wiley & Sons, Inc, 2003, vol. 44, ch. 27, pp. 547–557.

[18] S. Chaudhury, S. Lyskov, and J. Gray, "Pyrosetta: a script-based interface for implementing molecular modeling algorithms using rosetta," *Bioinformatics*, vol. 26, no. 5, pp. 689–691, 2010.

[19] L. D. L. Correa and M. Dorn, "A knowledge-based artificial bee colony algorithm for the 3-d protein structure prediction problem," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, July 2018, pp. 1–8.

[20] J. Schaarschmidt, B. Monastyrskyy, A. Kryshtafovych, and A. M. Bonvin, "Assessment of contact predictions in casp12: Co-evolution and deep learning coming of age," *Proteins: Struct. Funct. Bioinf.*, vol. 86, pp. 51–66, 2018.

[21] L. de Lima Corrêa and M. Dorn, "A multi-objective swarm-based algorithm for the prediction of protein structures," in *International Conference on Computational Science*. Springer, 2019, pp. 101–115.

[22] D. E. Kim, F. DiMaio, R. Yu-Ruei Wang, Y. Song, and D. Baker, "One contact for every twelve residues allows robust and accurate topology-level protein structure modeling," *Proteins: Struct. Funct. Bioinf.*, vol. 82, pp. 208–218, 2014.

[23] D. T. Jones, T. Singh, T. Kosciolek, and S. Tetchner, "Metapsicov: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins," *Bioinformatics*, vol. 31, no. 7, pp. 999–1006, 2014.

[24] B. Borguesan, M. B. e Silva, B. Grisci, M. Inostroza-Ponta, and M. Dorn, "APL: An angle probability list to improve knowledge-based metaheuristics for the three-dimensional protein structure prediction," *Comput. Biol. Chem.*, vol. 59, pp. 142–157, 2015.

[25] J. Dréo, A. Petrowski, P. Siarry, and E. Taillard, *Metaheuristics for hard optimization: methods and case studies*, 1st ed. USA: Springer Science & Business Media, 2006.

[26] L. de Lima Corrêa and M. Dorn, "Multi-agent systems in three-dimensional protein structure prediction," in *Multi-Agent-Based Simulations Applied to Biological and Environmental Systems*. IGI Global, 2017, pp. 241–278.

[27] A. G. Bedeian and K. W. Mossholder, "On the use of the coefficient of variation as a measure of diversity," *Organ. Res. Methods*, vol. 3, no. 3, pp. 285–297, 2000.

[28] M. Y. Lobanov, N. Bogatyreva, and O. Galzitskaya, "Radius of gyration as an indicator of protein structure compactness," *J. Mol. Biol.*, vol. 42, no. 4, pp. 623–628, 2008.

[29] W. L. DeLano, "The pymol molecular graphics system," 2002.