A Knowledge Based Self-Adaptive Differential Evolution Algorithm for Protein Structure Prediction

Pedro H. Narloch[1][0000000281429399] and Márcio Dorn[1][0000000185343480]

Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
mdorn@inf.ufrgs.br

Abstract. Tertiary protein structure prediction is one of the most challenging problems in Structural Bioinformatics, and it is a NP-Complete problem in computational complexity theory. The complexity is related to the significant number of possible conformations a single protein can assume. Metaheuristics became useful algorithms to find feasible solutions in viable computational time since exact algorithms are not capable. However, these stochastic methods are highly-dependent from parameter tuning for finding the balance between exploitation (local search refinement) and exploration (global exploratory search) capabilities. Thus, self-adaptive techniques were created to handle the parameter definition task, since it is time-consuming. In this paper, we enhance the Self-Adaptive Differential Evolution with problem-domain knowledge provided by the angle probability list approach, comparing it with every single mutation we used to compose our set of mutation operators. Moreover, a population diversity metric is used to analyze the behavior of each one of them. The proposed method was tested with ten protein sequences with different folding patterns. Results obtained showed that the self-adaptive mechanism has a better balance between the search capabilities, providing better results in regarding root mean square deviation and potential energy than the non-adaptive single-mutation methods.

Keywords: Protein Structure Prediction · Self-Adaptive Differential Evolution · Structural Bioinformatics · Knowledge-based Methods.

1 Introduction

Proteins are macro-molecules composed by a sequence of amino acids, assuming different shapes accordingly to this sequence and environment conditions [1]. The three-dimensional structural conformation of a protein is related to its biological function, where any modification might influence the protein's biological function [26]. Thus, the determination of these structures is significant to understanding proteins role performed inside a cell [9]. Nowadays, the determination of three-dimensional structures is through experimental methods such as X-ray crystallography and Nuclear Magnetic Resonance. However, these experimental strategies are time-consuming and expensive [12]. In light of the

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

importance of these molecules and limitations of experimental methods, computational strategies became interesting approaches to reduce costs and the difference between sequenced and determined structures. However, the determination of three-dimensional protein structures is classified, in computational complexity theory, as an NP-hard problem [15] due to the explosive of possible shapes a protein can assume, making impossible the use of exact methods to solve the problem. In light of the complexity of the Protein Structure Prediction (PSP) problem, metaheuristics became attractive to finding feasible solutions for one of the most challenging problems in Structural Bioinformatics [9], although these techniques do not guarantee the finding of optimal solution [13]. There are three steps needed to build a possible solver for the protein structure prediction, (i) the computational representation of proteins; (ii) a scoring method to measure the molecule's free energy; and (iii) a search method to explore the conformational search space [12]. Different metaheuristics have been used in many NP-hard problems but, the Differential Evolution [24] (DE) is one of the most effective search strategy for complex problems [11] in a vast type of problems, including PSP [18][19][20].

Besides the capacity of finding good solutions for NP-Complete problems that different metaheuristics have, they are very dependent on the balance between two search characteristics: the *exploitation* and the *exploration* [10]. This balance helps the algorithm to avoid local optima, prevent the premature convergence, and ensure the neighborhood exploitation for better final solutions. This balance can be affected by tuning parameters and modifying different operators, but this is not a trivial task. In this way, we propose the use of a Self-Adaptive Differential Evolution (SaDE) [22] in the PSP problem, since its adaptive mechanisms tend to preserve the balance between exploration and exploitation capabilities during the search process. As the PSP be a complex problem, we use the Angle Probability List [5] (APL), a valuable source of problem-domain data, to enhance the algorithm. Moreover, we also use a populational diversity metric [10] to monitor the SaDE behavior during the search process, comparing it with four mutation operators that compose the set used in the self-adaptive version. Some interesting convergence behaviors were observed as well as good results for the problem. The next sections in this paper are organized as follows. Section 2 presents the concepts used in this works such as the problem formulation, the SaDE algorithm, APL construction, and related works. The proposed method is described in Section 3. In Section 4 the results obtained by the different approaches are discussed. Conclusions and future works are given in Section 5.

2 Preliminaries

2.1 Three-Dimensional Protein Structure Prediction

A protein molecule is formed by a linear sequence of amino acids (primary structure). The thermodynamic hypothesis of Anfinsen [1] states that protein's folding depends on its primary structure. The native functional conformation of a protein molecule coincides with its lowest free energy conformation. Over the years,

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

different computational efforts were made in the PSP problem, creating energy functions, proteins representation, and search mechanisms to simulate the folding process [12]. However, as proteins are complex molecules, the definition of each of these three components is not a trivial task. The computational representation of proteins can vary, from the most simple ones such as two-dimensional *lattice* models [4] to the full-atom model in a three-dimensional space. The trade-off among these different representations is related to the computational cost versus real protein representation. One adequate way to computationally represent these complex molecules is by their rotational angles, known as dihedral angles, maintaining the closeness of real systems and reducing the computational complexity of its representation.

The dihedral angles of proteins are present in chemical bonds among the atoms that compose the molecule. The amino acids present in the protein's primary structure are chained together by a chemical bond known as a peptide bond. In general, all amino acids found in proteins have the same basic structure, with an amino-group N, the central carbon atom C_α, a carboxyl-group C, and four hydrogens. The difference among the 20 known amino acids is in their side-chain atoms. When bonding two amino acids, the peptide bond is formed by the C-N interaction, forming a planar angle known as ω. The ϕ angle represents the rotation around the N-C_α and ψ the rotation angle that rotates around C_α-C. These two angles (ϕ, ψ) are free to rotate in the space, varying from $-180°$ to $+180°$. Due to this fact, there is an explosion of possible conformation a protein can assume since each amino acid's backbone is composed of two free rotational and one planar angle. Beyond that, there are the side-chain angles noted as χ-angles, and their number varies from 0 to 4 accordingly to the amino acid type. The values of these rotation angles modify the position of different atoms along the whole protein structure, forming different structural patterns (secondary structure). The most stable and important secondary structures present in protein's structures are the α-helix and β-sheets. Another type of secondary structure is the β-turn, composed of short segments and generally responsible for connecting two β-strands. There are structures responsible for connecting different secondary structures, known as coils. In this way, ones can computationally represent a protein as a sequence of dihedral angles, where each set of angles serve as an amino acid. It is possible to imagine that as the size of a protein (quantity of amino acids in its primary structure) increases, the problem dimension grows as well.

The physicochemical interactions among the atoms should be considered to determine the correct orientation of them. In this way, different energy functions were proposed to simulate proteins molecular mechanics [3]. Prediction methods use a potential energy function to describe the search space, which the minimum global energy represents the native conformation of the protein. The *Rosetta energy function* [23] is one of the popular scoring tools for all-atom energy determination, and it is used in this paper. The Equation 1 presents the different components this energy function considers.

$$E_{Rosetta} = \begin{cases} E_{physics-based} + E_{inter-electrostatic} \\ +E_{H-bonds} + E_{knowledge-based} + E_{AA} \end{cases} \quad (1)$$

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

where $E_{physics-based}$ calculates the 6-12 Lennard-Jones interactions and Solvatation potential approximation, $E_{inter-electrostatic}$ stands for inter-atomic electrostatic interactions and $E_{H-bonds}$ hydrogen-bond potentials. In $E_{knowledge-based}$ the terms are combined with knowledge-based potentials while the free energy of amino acids in the unfolded state is in E_{AA} term.

Angle Probability List: Over the years different methods have been proposed for the PSP problem. These methods can be classified in four classes [12]. The fold recognition and comparative modeling are two classes of methods that strictly depends on existing structures to predict the structure of another protein. Besides their efficiency, they can not find new folds of proteins. In *first principles prediction without database information*, known as *ab initio*, the folding process uses only the amino acid as information for finding the lowest energy in the energy space, making possible the prediction of new folding patterns. However, methods purely *ab initio* have some limitations due to the size of the conformational search space [12]. In this work, we use a variation of the *ab initio* class, the *first principles with database information*. In this way, adding problem-domain knowledge to enhance the search mechanism, better structures are found, and it does not preclude the finding of new folding patterns.

As amino acids can assume different torsion angle values depending on their secondary structure [17], it is worth to consider these occurrences as information to reduce the search space while enhancing algorithms with better search capabilities. In light of these facts, the *Angle Probability List*, APL, was proposed in [5] based on the conformational preferences of amino acids based on their secondary structures. The data was retrieved from the Protein Data Bank (PDB) [2], considering only high-quality information. To compose this database, a set of 11,130 structures with resolution $\leq 2.5\mathring{A}$ was used. The APL was built based in a histogram matrix of $[-180, 180] \times [-180, 180]$ for each amino acid and secondary structure. To generate the APLs, a web tool known as NIAS [1] (*Neighbors Influence of Amino acids and Secondary structures*) was used [6].

Self-Adaptive Differential Evolution: The Differential Evolution (DE) algorithm was proposed initially by Storn and Price [24] and since then it has been one of the most efficient metaheuristics in different areas [11]. The DE is a populational-based evolutionary algorithm which depends on three parameters, the crossover rate (CR), a mutation factor (F) and the size of the population (NP). In the SaDE [22] version, parameters CR and F are modified by the algorithm instead of pre-fixed values for the whole optimization process. This strategy is interesting since the parameter fine-tuning is a time-consuming task. Another important fact is that there is not a global parameter value that might be the optimum parameter for all problems.

As the F factor be related to the convergence speed, in SaDE algorithm the F parameter assume random values in the range of $[0, 2]$, with a normal distribution of mean 0.5 and standard deviation of 0.3. In this way, the global (large F values) and local (low F values) search abilities are maintained during the whole optimization process. The CR parameter is changed along the evolutionary pro-

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

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

cess, starting with a random mean value of 0.5 (CRm) and a standard deviation of 0.1. The CRm is adapted during the optimization process based on its success rate. Furthermore, the SaDE also adapts the mutation mechanism used for creating new individuals. In classical DE algorithm, only one mutation mechanism is employed during the whole optimization process. The first SaDE approach proposed the usage of two different mutation mechanisms, with different exploration and exploitation capabilities. To chose the method to be employed, a learning stage is applied during some generations before the real optimization process. In this way, a probability of occurrence is associated with a mutation mechanism accordingly its success and failure rate.

Related Works: Some of the most well-known search algorithms used in the PSP problem are Genetic Algorithms (GA), Differential Evolution (DE), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO) and many others. The DE behavior was previously analyzed in [18] and [19], where different mutation strategies were employed to increase the diversity capabilities of the algorithm. As the author used the diversity metric, it is possible to notice that the diversity maintenance is a key factor to avoid local optima solutions and, consequently, the premature convergence. A self-adaptive multi-objective DE was proposed in [25], showing the importance of how self-adaptive strategies could be interesting to the PSP problem. The Self-Adaptive Differential Evolution was employed by [20] with two sources of knowledge: the APL and the Structure Pattern List (SPL). In this version, authors demonstrated how important it is to combine problem-domain knowledge with the SaDE algorithm. Besides the contribution of using APL and SPL as a source of structural information, the authors have not analyzed each mutation operator separately, either the algorithm behavior regarding convergence and diversity maintenance, creating a gap in the application. Besides some works have already used APL as a source of information [5][7][9], none of them have used some self-adaptive mechanism or are concerned about the behavior of the algorithms regarding diversity maintenance. Thus, in our approach, we close this gap using a diversity index to monitor and analyze the behavior of each mutation operator and a self-adaptive version of the DE algorithm combined with information provided by the APL. Also, our application uses different mutation operators from [20] based on the exploration and exploitation capabilities of each mutation strategy.

3 Material and Methods

There are three essential components needed to create a PSP predictor: (i) a way to computationally represents the protein structure; (ii) a scoring function to evaluate the protein's potential energy; and (iii) a search strategy to explore the protein's conformational search space and find feasible structures. The main contribution of this work is related to the (iii) search strategy, providing a populational convergence analysis of each mutation mechanism used in a knowledge-based SaDE algorithm for the PSP problem.

Protein's Representation and Scoring Function: In this work, we represented a protein molecule as a set of torsion angles. Each possible solution

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

assumes $2N$ dimensions, where N is the length of the protein's primary structure. Therefore, this set of angles modifies the cartesian coordinates of protein's atoms to do the energy evaluation of the molecule. As we use the PyRosetta [8], a well-known interface to Python-based Rosetta energy function interface [23], we opted to reduce the search space optimizing only the protein backbone torsion angles (ϕ and ψ) without losing the molecule's characteristics. In light of preserving well-formed secondary structures, we used the PyRosetta to identify secondary structures using `DSSP` implementation [16] and considering it as an additional term in the *score3* energy function as shown by Equation 2.

$$E_{total} = E_{score3} + E_{SS} \tag{2}$$

Another important metric to evaluate a possible solution is the *Root Mean Square Deviation* (`RMSD`), which compares the distance, in angstroms, among the atoms in two structures. In this work, the `RMSD` is used to compare the final solution with the already known experimental structure. Equation 3 displays the $\texttt{RMSD}_\alpha$ metric, which compares the backbone between two structures.

$$\texttt{RMSD}(a,b) = \sqrt{\frac{\sum_{i=1}^{n} \mid r_{ai} - r_{bi} \mid^2}{n}} \tag{3}$$

where r_{ai} and r_{bi} are the ith atoms in a group of n atoms from structures a and b. The closer `RMSD` is from 0Å more similar are the structures.

Search Strategy: In any metaheuristic, the adjustment of parameters is important but not a trivial task since they affect the quality of possible solutions [14]. In order to sidestep the time-consuming task of parameter tuning, different self-adaptive strategies were proposed [21]. In this work we combine the `SaDE` [22] approach with the `APL` knowledge-database considering the high-quality information it provides [5][7][9]. As far as we know, the only `SaDE` application that used some kind of structural information was proposed in [20]. Differently from [22], we have used four `DE` mutation mechanisms (Table 1), which are also different from the used in [20]. We took in consideration the exploratory ($\texttt{DE}_{rand/1/bin}$ and $\texttt{DE}_{curr-to-rand}$) and exploitative ($\texttt{DE}_{best/1/bin}$ and $\texttt{DE}_{curr-to-bes}$) capabilities they provide to compose the set of mutation mechanisms that `SaDE` can choose. The Algorithm 1 shows the how we have structured our approach. The "learning stage" uses the same structure but with few numbers of generations to set the initial probability rates of each mutation strategy and CRm.

Moreover, we use a diversity measure (Equation 4) to monitor the algorithm behavior during the optimization process. This metric takes into consideration the individual dimensions instead of the fitness, making possible to verify if the population has lost its diversity. This index was proposed in [10] for continuous-domain problems. The index ranges from $[0, 1]$, where 1 is the maximum diversity in the population and 0 the full convergence of the population to a single solution.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

Approach	Equation
$\text{DE}_{best/1/bin}$	$\boldsymbol{v}_i^{g+1} = \boldsymbol{x}_{best}^g + F \cdot (\boldsymbol{x}_{r2}^g - \boldsymbol{x}_{r3}^g)$
$\text{DE}_{rand/1/bin}$	$\boldsymbol{v}_i^{g+1} = \boldsymbol{x}_{r1}^g + F \cdot (\boldsymbol{x}_{r2}^g - \boldsymbol{x}_{r3}^g)$
$\text{DE}_{curr-to-rand}$	$\boldsymbol{v}_i^{g+1} = \boldsymbol{x}_i^g + F1 \cdot (\boldsymbol{x}_{r1}^g - \boldsymbol{x}_i^g) + F2 \cdot (\boldsymbol{x}_{r2}^g - \boldsymbol{x}_{r3}^g)$
$\text{DE}_{curr-to-best}$	$\boldsymbol{v}_i^{g+1} = \boldsymbol{x}_i^g + F1 \cdot (\boldsymbol{x}_{best}^g - \boldsymbol{x}_i^g) + F2 \cdot (\boldsymbol{x}_{r2}^g - \boldsymbol{x}_{r3}^g)$

Table 1: Classical mutation strategies in DE.

$$GDM = \frac{\sum_{i=1}^{N-1} ln\left(1 + \min_{j[i+1,N]} \frac{1}{D}\sqrt{\sum_{k=1}^{D}(x_{i,k} - x_{j,k})^2}\right)}{NMDF} \tag{4}$$

where D represents the dimensionality of the solution vector, N is the population size and x the individual (the solution vector). The NMDF is a normalization factor which corresponds to the maximum diversity value so far.

Algorithm 1 Self-Adaptive Differential Evolution with APL

Data: NP
Result: The best individual in population
Generate initial population with NP individuals based on APL
while $g \leq$ *number of generations* **do**
  F ← norm(0.5, 0.3)
  if *past 25 generations* **then**
    CRm ← update based on the success rate of previous CR values.
  end
  for *each i individual in population* **do**
    mStrategy ← random(0,1) //Probability to choose the mutation strategy
    modifies the individual $u_{i,g}$ with the mutation strategy accordingly to mStrategy
    if $\boldsymbol{u}_{i,fitness} \leq \boldsymbol{x}_{i,fitness}$ **then**
      add $\boldsymbol{u_i}$ in the offspring
    else
      add $\boldsymbol{x_i}$ in the offspring
    end
  end
  update the mutation probabilities based on their success rate
  population ← offspring
  g ← g +1
end

4 Experiments and Analysis

The algorithms was ran 30 times in five different DE configurations: $DE_{rand/1/bin}$, $DE_{best/1/bin}$, $DE_{curr-to-rand}$, $DE_{curr-to-best}$ and $DE_{Self-Adaptive}$. For the four

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

non-adaptive versions of DE, we used the parameter CR as 1 and F as 0.5, while $DE_{Self-Adaptive}$ the parameters are initialized with 0.5 for CRm and 0.5 for F. One million of fitness evaluations were done in each run, corresponding to 10 thousand generations in total. To keep a fair comparison among all different versions, the initial population for each DE configuration is the same, avoiding that one mechanism starts with a better population than others. Achieved results are present in Table 2, by protein and DE version. Tests were performed in an *Intel Xeon E5-2650V4 30 MB, 4 CPUs, 2.2Ghz, 96 cores/threads, 128G of RAM, and 4TB* in disk space. To test our approach we used 9 proteins based on literature works that can be foun at the PDB. The PDB ID are: 1AB1 (46 amino acids), 1ACW (29 amino acids), 1CRN (46 amino acids), 1ENH (54 amino acids), 1ROP (63 amino acids), 1UTG (70 amino acids), 1ZDD (35 amino acids), 2MR9 (44 amino acids), and 2MTW (20 amino acids).

In order to compare the SaDE approach with the other 4 DE variations, we have applied the *Wilcoxon Signed Rank Test* (Table 2 - 3rd column), where *p-values* lower than 0.05 indicates that there is statistical relevance. It is possible to notice that $\text{DE}_{Self-Adaptive}$ got relevant results in 5 of 9 cases, and better average energy in 8 of them. In the other 3 cases, SaDE showed equivalence with $\text{DE}_{curr-to-rand}$ (1ENH and 1UTG) and $\text{DE}_{rand/1/bin}$ (1ROP), getting worst results only for 2MTW, meaning that the Self-Adaptive approach is better, or at least equivalent, to the non-adaptive version of DE using only one mutation mechanism in 8 of 9 cases. The convergence analysis are presented by two proteins (1ROP and 1UTG) in Figure 1 and Figure 2. For other proteins, the patterns are quite similar, changing accordingly with the dimensionality each protein presents. For 1ROP protein (Figure 1) it is possible to notice that $\text{DE}_{Self-Adaptive}$ better explore the search space during the optimization process, leading to better energy values, and avoiding premature convergence as observed by $\text{DE}_{best/1/bin}$. A similar analysis can be done in the 1UTG protein's optimization process (Figure 2), where the diversity index from $\text{DE}_{Self-Adaptive}$ is significant even in the end of the optimization process. This behavior shows that it is possible to keep optimizing the search space for even better solutions.

PDB	Strategy	Energy	$p-value$
1AB1	$\text{DE}_{rand/1/bin}$	$-98.00(-75.48 \pm 9.54)$	0.00
	$\text{DE}_{best/1/bin}$	$-152.24(-95.32 \pm 18.48)$	0.00
	$\text{DE}_{curr-to-rand}$	$-169.14(-109.07 \pm 17.29)$	0.00
	$\text{DE}_{curr-to-best}$	$-158.14(-122.57 \pm 15.64)$	0.00
	$\mathbf{DE_{Self-Adaptive}}$	$\mathbf{-184.62(-157.08 \pm 20.37)}$	–
1ACW	$\text{DE}_{rand/1/bin}$	$-148.22(-25.17 \pm 41.97)$	0.00
	$\text{DE}_{best/1/bin}$	$-133.85(-88.22 \pm 39.33)$	0.00
	$\text{DE}_{curr-to-rand}$	$-135.75(-63.13 \pm 24.92)$	0.00
	$\text{DE}_{curr-to-best}$	$-160.84(-111.85 \pm 26.69)$	0.00
	$\mathbf{DE_{Self-Adaptive}}$	$\mathbf{-203.31(-161.35 \pm 20.04)}$	–

Continued on next page

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

Table 2 – *Continued from previous page*

PDB	Strategy	Energy	$p-value$
1CRN	$DE_{rand/1/bin}$	$-95.03(-72.76 \pm 6.13)$	0.00
	$DE_{best/1/bin}$	$-136.18(-93.92 \pm 16.06)$	0.00
	$DE_{curr-to-rand}$	$-188.41(-113.55 \pm 23.59)$	0.00
	$DE_{curr-to-best}$	$-173.95(-129.20 \pm 23.17)$	0.00
	$\mathbf{DE_{Self-Adaptive}}$	$\mathbf{-185.67(-154.13 \pm 18.55)}$	–
1ENH	$DE_{rand/1/bin}$	$-343.13(-334.83 \pm 3.08)$	0.00
	$DE_{best/1/bin}$	$-364.38(-348.84 \pm 7.92)$	0.00
	$\mathbf{DE_{curr-to-rand}}$	$\mathbf{-376.11(-363.21 \pm 10.90)}$	0.20
	$DE_{curr-to-best}$	$-368.94(-359.37 \pm 5.06)$	0.00
	$\mathbf{DE_{Self-Adaptive}}$	$\mathbf{-375.94(-367.26 \pm 4.35)}$	–
1ROP	$\mathbf{DE_{rand/1/bin}}$	$\mathbf{-498.18(-485.32 \pm 6.59)}$	0.28
	$DE_{best/1/bin}$	$-471.52(-458.66 \pm 6.13)$	0.00
	$DE_{curr-to-rand}$	$-484.88(-475.80 \pm 3.14)$	0.00
	$DE_{curr-to-best}$	$-477.11(-468.65 \pm 4.64)$	0.00
	$\mathbf{DE_{Self-Adaptive}}$	$\mathbf{-507.13(-488.14 \pm 8.78)}$	–
1UTG	$DE_{rand/1/bin}$	$-514.55(-487.69 \pm 10.24)$	0.00
	$DE_{best/1/bin}$	$-516.13(-497.01 \pm 9.29)$	0.00
	$\mathbf{DE_{curr-to-rand}}$	$\mathbf{-545.70(-533.13 \pm 8.03)}$	0.42
	$DE_{curr-to-best}$	$-536.09(-515.88 \pm 9.49)$	0.00
	$\mathbf{DE_{Self-Adaptive}}$	$\mathbf{-544.34(-534.29 \pm 5.72)}$	–
1ZDD	$DE_{rand/1/bin}$	$-233.00(-225.00 \pm 3.78)$	0.00
	$DE_{best/1/bin}$	$-232.28(-225.54 \pm 3.66)$	0.00
	$DE_{curr-to-rand}$	$-245.71(-236.38 \pm 4.22)$	0.00
	$DE_{curr-to-best}$	$-240.61(-231.89 \pm 4.05)$	0.00
	$\mathbf{DE_{Self-Adaptive}}$	$\mathbf{-245.49(-240.38 \pm 3.26)}$	–
2MR9	$DE_{rand/1/bin}$	$-287.20(-264.20 \pm 11.33)$	0.00
	$DE_{best/1/bin}$	$-282.84(-270.72 \pm 6.96)$	0.00
	$DE_{curr-to-rand}$	$-296.22(-289.38 \pm 3.28)$	0.02
	$DE_{curr-to-best}$	$-290.33(-283.44 \pm 4.76)$	0.00
	$\mathbf{DE_{Self-Adaptive}}$	$\mathbf{-299.87(-290.89 \pm 3.86)}$	–
2MTW	$\mathbf{DE_{rand/1/bin}}$	$\mathbf{-109.56(-102.87 \pm 3.45)}$	0.01
	$DE_{best/1/bin}$	$-95.02(-90.62 \pm 2.12)$	0.00
	$DE_{curr-to-rand}$	$-104.58(-98.74 \pm 2.88)$	0.01
	$DE_{curr-to-best}$	$-101.91(-94.70 \pm 2.53)$	0.00
	$DE_{Self-Adaptive}$	$-105.3(-100.66 \pm 2.13)$	–

Table 2: Results obtained by the 5 DE approaches. Bolded lines presents the approaches with best energy results accordingly to *Wilcoxon Signed Rank Test.*

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

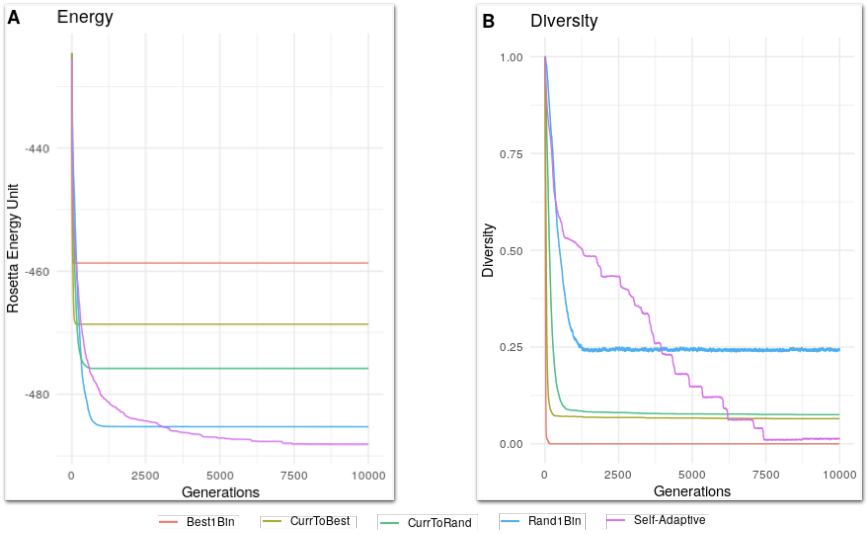


Fig. 1: PDB ID 1ROP convergence of energy and diversity for all five Differential Evolution versions. Both plots consider the average among all runs.

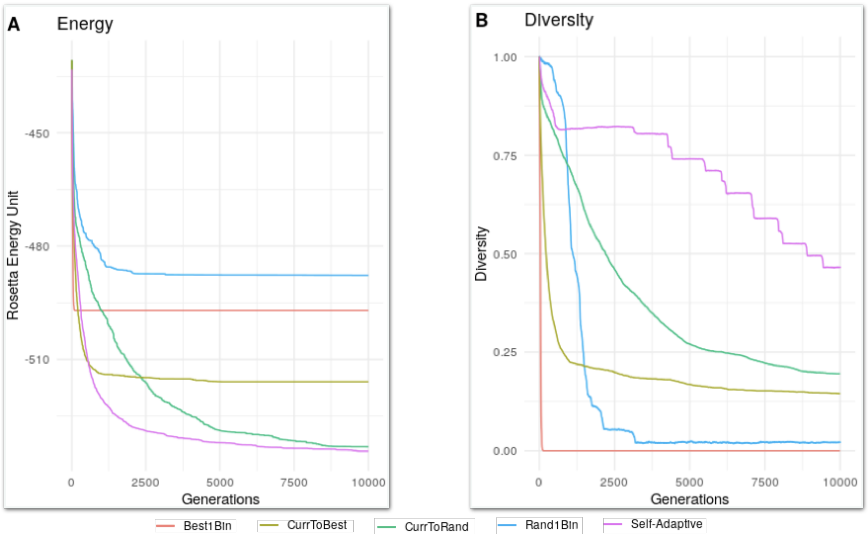
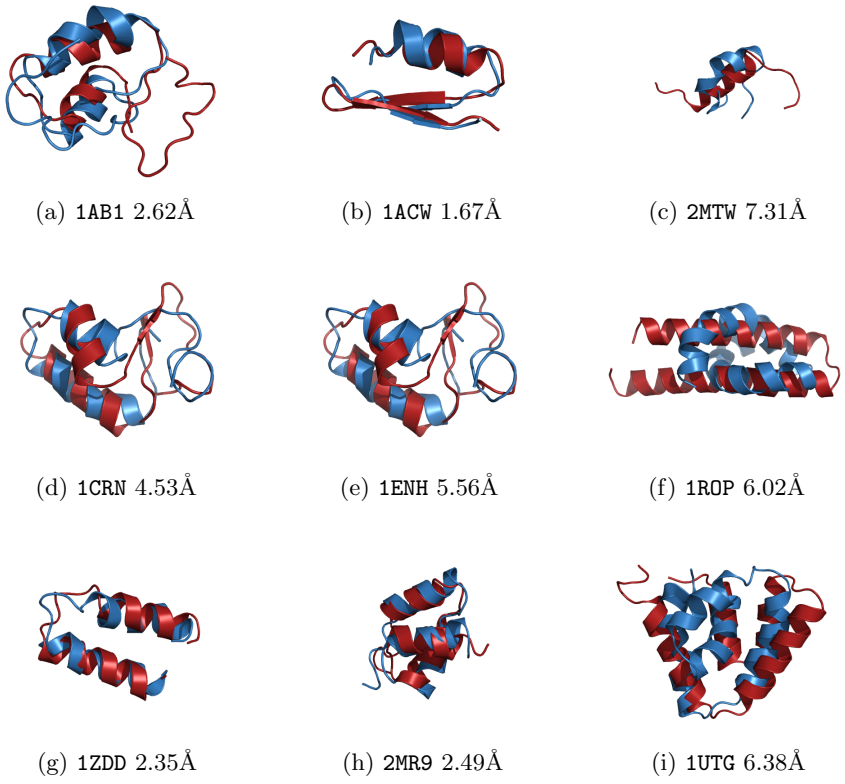


Fig. 2: PDB ID 1UTG convergence of energy and diversity for all five Differential Evolution versions. Both plots consider the average among all runs.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

This analysis shows that the combination of different mutation mechanisms during the optimization process can be beneficial to the balance between exploration and exploitation capabilities. It is possible to observe that elitist approaches ($\texttt{DE}_{best/1/bin}$ and $\texttt{DE}_{curr-to-best}$) are not so good when used alone during the whole process, but they are useful in small portions of generations. Since the determination of **when** apply this type of technique, the Self-Adaptive mechanism can decide by itself when to use each mutation operator. Also, the self-adaptive mechanism used adapts the mutation and crossover factors (F and CR), which might contribute to better search space exploration. It is noteworthy that each protein configures a different search space. Hence, the parameter setting for one protein might not be better for every other protein. The same assumption can be used for mutation operators. Final conformations are compared with the experimental ones and reported in Fig. 3



(a) 1AB1 2.62Å (b) 1ACW 1.67Å (c) 2MTW 7.31Å

(d) 1CRN 4.53Å (e) 1ENH 5.56Å (f) 1ROP 6.02Å

(g) 1ZDD 2.35Å (h) 2MR9 2.49Å (i) 1UTG 6.38Å

Fig. 3: Cartoon representation of experimental structures (red) compared with lowest energy solutions (blue) found by SaDE version.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

It is possible to notice that $\texttt{DE}_{Self-Adaptive}$ achieved the better results in terms of Energy and RMSD when compared with the other approaches . Of course, it is needed further investigations to improve the conformational search method, helping the algorithm to reach more similar structures, with lower energy and RMSD. It is important to realize that the RMSD values should decrease within the energy values, but as the energy functions are computational approximations, and the search space has multimodal characteristics, it is possible to have conformations with higher energy values but with lower RMSD values. However, it is expected that if the minimum global energy is found, the RMSD might be 0, finding the correct structure.

5 Conclusion

As shown in many works in literature, the PSP problem still an open issue in Bioinformatics that can contribute for life-sciences. Besides the significant advances in the problem, it is still needed advances in better search methods for prediction of proteins. As proteins have different characteristics among them, and metaheuristics are very sensitive in the use of parameters and operators, the parameter tuning and mechanism choice is not a trivial task, if not an impossible one. Thus, we have used a self-adaptive version of the differential evolution ($\text{DE}_{Self-Adaptive}$) algorithm to solve the PSP problem, combining four well-known mutation mechanisms not yet combined for this problem. Moreover, we have used a diversity measure to analyze the behavior of each mechanism and the combination of all of them in the SaDE algorithm, something not yet explored in the literature. Accordingly to the convergence graphs and diversity measure, it was possible to verify that elitist approaches ($\text{DE}_{best/1/bin}$ and $\text{DE}_{curr-to-best}$) quickly loss the populational diversity while the random ones ($\text{DE}_{rand/1/bin}$ and $\text{DE}_{curr-to-rand}$) have slower convergence. The combination of them in a self-adaptive model seems to contribute to a better balance between exploitation and exploration mechanisms, allowing the algorithm to find better solutions.

As the problem of predicting tertiary structures of proteins being complex, it is imminent the usage of some problem-domain knowledge. In light of this fact, we have used the information of the conformational preferences of amino acids provided by the APL. The data supplied by the APL have been shown beneficial in different algorithms, such as GAs and PSO. The results obtained in our work are not only interesting regarding problem-solving, but also in algorithm behavior analysis. The $\texttt{DE}_{Self-Adaptive}$ got better results in 5 of 9 cases with 95% of confidence accordingly to the *Wilcoxon Signed Rank Test* and being equivalent in other 3 cases. Also, the diversity measure showed that self-adaptive mechanisms enhanced the algorithm capabilities for better exploration of the search space and, consequently, better energy results. Although the SaDE algorithm was already used in [20] with attached problem-domain knowledge, an analysis of each mutation mechanism was not found, neither the energy values or any type of convergence trace was done. In this way, the present work closed this gap, pro-

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

viding the opportunity to do further investigations of self-adaptive algorithms using APL as a knowledge database to enhance the algorithm capabilities.

For future works, it is intended to expand the usage of APL in different ways, not only in the initial population. Also, it would be interesting to add more DE mechanisms, comparing their behavior with specific metrics (such as the diversity measurement), and how they contribute to the self-adaptive algorithm for better search capabilities. It is important to do better investigations about the energy functions, verifying the possibility of multiobjective problem formulation as already seen in other PSP predictors that used different energy functions to guide the search mechanism.

Acknowledgements

This work was supported by grants from FAPERGS [*16/2551-0000520-6*], MCT/CNPq [*311022/2015-4; 311611/2018-4*], CAPES-STIC AMSUD [*88887.135130/2017-01*] - Brazil, Alexander von Humboldt-Stiftung (AvH) [*BRA 1190826 HFST* CAPES-P] - Germany. This study was financed in part by the Coordenacão de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001.

References

1. Anfinsen, C.B.: Principles that Govern the Folding of Protein Chains. Science **181**(4096), 223–230 (7 1973)
2. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic Acids Res **28**, 235–242 (2000)
3. Boas, F.E., Harbury, P.B.: Potential energy functions for protein design. Current Opinion in Structural Biology **17**(2), 199–204 (2007)
4. Bonneau, R., Baker, D.: Ab initio protein structure prediction: Progress and prospects. Annual Review of Biophysics and Biomolecular Structure **30**(1), 173–189 (2001)
5. Borguesan, B., E Silva, M.B., Grisci, B., Inostroza-Ponta, M., Dorn, M.: APL: An angle probability list to improve knowledge-based metaheuristics for the three-dimensional protein structure prediction. Computational Biology and Chemistry **59**, 142–157 (2015)
6. Borguesan, B., Inostroza-Ponta, M., Dorn, M.: NIAS-Server: Neighbors Influence of Amino acids and Secondary Structures in Proteins. Journal of Computational Biology **24**(3), 255–265 (2017)
7. Borguesan, B., Narloch, P.H., Inostroza-Ponta, M., Dorn, M.: A Genetic Algorithm Based on Restricted Tournament Selection for the 3D-PSP Problem. In: 2018 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8. IEEE (jul 2018)
8. Chaudhury, S., Lyskov, S., Gray, J.J.: PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. Bioinformatics **26**(5), 689–691 (3 2010)
9. Corrêa, L.d.L., Borguesan, B., Krause, M.J., Dorn, M.: Three-dimensional protein structure prediction based on memetic algorithms. Computers and Operations Research **91**, 160–177 (2018)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7

10. Corriveau, G., Guilbault, R., Tahan, A., Sabourin, R.: Review of phenotypic diversity formulations for diagnostic tool. Applied Soft Computing Journal **13**(1), 9–26 (2013)
11. Das, S., Mullick, S.S., Suganthan, P.N.: Recent advances in differential evolution-An updated survey. Swarm and Evolutionary Computation **27**, 1–30 (2016)
12. Dorn, M., E Silva, M.B., Buriol, L.S., Lamb, L.C.: Three-dimensional protein structure prediction: Methods and computational strategies. Computational Biology and Chemistry **53**(PB), 251–276 (2014)
13. Du, K.l.: Search and Optimization by Metaheuristics Techniques and Algorithms Inspired by Nature
14. Eiben, E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms - evolutionary computation, ieee transactions on. October **3**(2), 124–141 (1999)
15. Guyeux, C., Côté, N.M.L., Bahi, J.M., Bienie, W.: Is Protein Folding Problem Really a NP-Complete One ? First Investigations. Journal of Bioinformatics and Computational Biology **12**(01) (feb 2014)
16. Kabsch, W., Sander, C.: Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. Biopolymers **22**(12), 2577–2637 (12 1983)
17. Ligabue-Braun, R., Borguesan, B., Verli, H., Krause, M.J., Dorn, M.: Everyone Is a Protagonist: Residue Conformational Preferences in High-Resolution Protein Structures. Journal of Computational Biology (4) (2017)
18. Narloch, P., Parpinelli, R.: Diversification strategies in differential evolution algorithm to solve the protein structure prediction problem, vol. 557 (2017)
19. Narloch, P., Parpinelli, R.: The protein structure prediction problem approached by a cascade differential evolution algorithm using ROSETTA. In: Proceedings - 2017 Brazilian Conference on Intelligent Systems, BRACIS 2017 (2018)
20. Oliveira, M., Borguesan, B., Dorn, M.: SADE-SPL: A Self-Adapting Differential Evolution algorithm with a loop Structure Pattern Library for the PSP problem. In: 2017 IEEE Congress on Evolutionary Computation (CEC). pp. 1095–1102 (6 2017)
21. Parpinelli, R.S., Plichoski, G.F., Samuel, R., Narloch, P.H.: A review of techniques for on-line control of parameters in swarm intelligence and evolutionary computation algorithms. International Journal of Bio-inspired Computation (2018)
22. Qin, A., Suganthan, P.: Self-adaptive Differential Evolution Algorithm for Numerical Optimization. 2005 IEEE Congress on Evolutionary Computation **2**, 1785–1791 (2005)
23. Rohl, C.A., Strauss, C.E., Misura, K.M., Baker, D.: Protein Structure Prediction Using Rosetta. pp. 66–93 (2004)
24. Storn, R., Price, K.: Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization **11**(4), 341–359 (1997)
25. Venske, S.M., Gonçalves, R.A., Benelli, E.M., Delgado, M.R.: ADEMO/D: An adaptive differential evolution for protein structure prediction problem. Expert Systems with Applications **56**, 209–226 (2016)
26. Walsh, G.: Proteins: Biochemistry and Biotechnology. Wiley (2014)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_7