2013 Brazilian Conference on Intelligent Systems

A Multiobjective Algorithm for Protein Structure Prediction Using Adaptive Differential Evolution

Sandra M. Scós Venske
CPGEI
UTFPR
Curitiba, Brazil
sandravenske@gmail.com

Richard A. Gonçalves
Department of Computer Science
UNICENTRO
Guarapuava, Brazil
richard@unicentro.br

Elaine M. Benelli
Department of Biochemistry
and Molecular Biology - UFPR
Curitiba, Brazil
benelli@ufpr.br

Myriam R. Delgado
CPGEI/DAINF
UTFPR
Curitiba, Brazil
myriamdelg@utfpr.edu.br

***Abstract*—Protein Structure Prediction (PSP) is one of the most challenging problems in Bioinformatics research area. This paper models PSP as a multiobjective optimization problem and adopts Adaptive Differential Evolution for Multiobjective Problems (ADEMO/D) to minimize potential energies (bonded and non-bonded) providing final protein structures. ADEMO/D incorporates concepts of Multiobjective Evolutionary Algorithms based on Decomposition (MOEA/D) and mechanisms of mutation strategies adaptation. In this work the probability matching and extreme absolute reward methods are combined to adapt ADEMO/D to the PSP context. The DE mutation strategy is chosen from a candidate pool according to a probability that depends on its received reward. We test the behavior of the proposed method, considering the off-lattice model and *ab initio* approach for PSP, in *Met-Enkephalin* peptide and 1ZDD protein. The results point ADEMO/D as a competitive approach for potential energy values and conformation similarity metrics.**

Keywords*—*Multiobjective Optimization; Adaptive Differential Evolution; Probability Matching.

I. Introduction

Proteins are composed by one or more polypeptide chains, each one containing from several to hundreds or even thousands amino acids, and are responsible for many different biological functions. Three dimensional structures of proteins are often necessary to understand their functions at a molecular level [1]. Protein structure prediction (PSP) is one of the most challenging problems nowadays and an important Bioinformatics research topic. Other example are sequence analysis, protein function prediction, protein-protein interaction prediction and microarray analysis [2]. The American biochemist Christian Anfinsen established the foundation for PSP: all information needed to predict the native conformation is encoded in its amino acid sequence [3].

When a protein is in its folded state, its free energy conformation is the lowest one. Designing an energy minimization approach to protein structure prediction is very hard because proteins are very flexible; therefore, their potential conformation space is too large to be exhaustively explored. Despite the great number of researches, the solution to the PSP problem is still an open field [4].

Usually, the evaluation of the conformation of a protein is based on the estimates of two energies: bond atoms and non-bond atoms. Recent experimental research indicates that these interactions are in conflict [5], [6], [7], justifying a multiobjective formulation for the PSP.

Three-dimensional structures of some proteins have been determined experimentally, commonly, using X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy. Although very precise, these methods are slow and costly. Computational strategies have thus been developed to offer alternatives for solving PSP problems [8] and evolutionary methods appear as strong candidates. Differential Evolution (DE) [9] is one of these promising evolutionary methods.

This paper tests ADEMO/D - *Adaptive Differential Evolution for Multiobjective Optimization* in the PSP problem. We model PSP as an optimization problem aiming to minimize protein's bonded and non-bonded potential energies, using an evolutionary algorithm. The addressed PSP assumes an off-lattice model where a protein is represented as a chain of residues or groups of residues moving through continuous space. Furthermore, we consider the *ab initio* approach for PSP, that is a template-free modeling and is by now recognized as one of the most difficult problems in computational structural biology [10]. We use a modified version of the method proposed in [11] which achieved good results for a multiobjective benchmark. This set of characteristics turns this work a relevant contribution for the Bioinformatics and Multiobjective Optimization research areas.

This paper has been structured as follows. Section II presents the PSP problem. Basic concepts of multi-objective optimization and differential evolution are discussed in Section III. Section IV details the proposed approach. Section V presents the experiment results concerning two proteins and final remarks are presented in Section VI.

II. Protein Structure Prediction

The base of a protein is its *primary structure* (formed by amino acids chains) which determines its different chemical properties. Amino acids rearrange themselves creating local folds, mostly α-helices and β-strands, called *secondary structure* (2D) of the polypeptide chain. *Tertiary structure* (native) is the arrangement of secondary structure elements in three dimensional spaces. Proteins assume a three-dimensional shape which is usually responsible for their function [8]. Spontaneously, proteins fold into a unique three-dimensional structure determined by their primary structure.

978-0-7695-5092-3/13 $26.00 © 2013 IEEE
DOI 10.1109/BRACIS.2013.51

Computational strategies have come to assist the process of determining the three-dimensional structures of proteins, as an alternative to experimental methods (X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy), which are slow and costly. Some strategies deal with *knowledge-based models* like homology modeling and threading [8]. The *ab initio* strategies are used when no homology is available, thus protein structure is predicted from primary sequence without using any structural template [8]. In this work we follow the *ab initio* strategy with an off-lattice model, representing a protein as a chain of residues or groups of residues that fold in a continuous space [8].

A. CHARMM force field

Representing the protein energy as a function of its atomic coordinates is important to simulate the protein molecular dynamics. The *potential energy functions* (or force fields) return a value for the energy according to molecule conformations. *Ab initio* PSP problem aims to minimize these functions. There are many force fields available [12], [13]. In this work we use CHARMM (v.27) force field calculated as a function of internal (bonded) and external (interaction or non-bonded) terms. Additionally, the final conformation energy given by $E_{charmm} = E_b + E_{nb}$ is used as the decision maker criterion, i.e., the decision maker chooses the structure with the lowest E_{charmm} value. The bonded and non-bonded terms are formed by Eq. 1 and 2, respectively.

$$\begin{aligned} E_b = & \sum_{bounds} K_b(b - b_0)^2 + \sum_{UB} K_{UB}(S - S_0)^2 \\ & + \sum_{angles} K_\theta(\theta - \theta_0)^2 + \sum_{dihedrals} K_\Phi[1 + cos(n\Phi - \gamma)] \\ & + \sum_{impropers} K_{imp}(\varphi - \varphi_0)^2 \end{aligned} \tag{1}$$

The terms in Eq. 1 describe the geometry of the molecule and are represented by bond length, b; the valence angle, θ; the distance between atoms separated by two covalent bonds, S; the dihedral or torsion angle, Φ; the improper angle, φ; and the distance between atoms i and j, r_{ij} [8]. Others parameters include the bond force constant and equilibrium distance, K_b and b_0 , respectively; the valence angle force constant and equilibrium angle, K_θ, and θ_0; the Urey-Bradley force constant and equilibrium distance, K_{UB} and S_0; the dihedral angle force constant, multiplicity, and phase angle, K_Φ, n, and γ; and the improper force constant and equilibrium improper angle, K_{imp} and φ_0.

$$E_{nb} = \sum_{nb} \varepsilon_{ij} \left[\left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{12} - \left(\frac{R_{min_{ij}}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{e r_{ij}} \tag{2}$$

The terms in Eq. 2 are the parameters that describe the interactions between atoms i and j and include the partial atomic charges, q_i, the Lennard Jones well-depth, ε_{ij}, and minimum interaction radius, $R_{min_{ij}}$, used to calculate the van der Waals interactions.

B. Constraints of the addressed PSP problem

This work adopts a model based on off-lattice and internal coordinates representation - the torsion angles - with backbone and sidechain torsion angles to model proteins. Each residue type has a pre-established number of torsion angles to reach the protein conformation. The backbone of each residue is represented by 3 dihedral angles: ϕ, ψ, ω. The sidechains are represented by χ_i angles. The number of χ angles depends on the residue type (see Table I). Aiming to reduce the search space, angles are restricted to specific ranges, based on the full DSSP 8-class classification [14]. Table II is based on secondary structures present in proteins and provides information regarding constraints in the torsion angles. The secondary structure constraints for peptides are predicted in this work using [15].

TABLE I. NUMBER OF χ ANGLES IN EACH RESIDUE.

Residue	Number of χ angles
GLY, ALA, PRO	main chain
SER, CYS, THR, VAL	χ_1
ILE, LEU, ASP, ASN, PHE, TYR, TRP	χ_1, χ_2
MET, GLU, GLN	χ_1, χ_2, χ_3
LYS, ARG	χ_1, χ_2, χ_3, χ_4

TABLE II. SECONDARY STRUCTURE CONSTRAINTS REGIONS.

Secondary structure	ϕ	ψ
H (α-helix)	[-67°, -47°]	[-57°, -37°]
B (β-bridge)	[-130°, -110°]	[110°, 130°]
E (β-strand)	[-130°, -110°]	[110°, 130°]
G (3-10-helix)	[-59°, -39°]	[-36°, 16°]
I (pi-helix)	[-67°, -47°]	[-80°, -60°]
T (turn)	[-180°, 180°]	[-180°, 180°]
S (bend)	[-180°, 180°]	[-180°, 180°]
U (undefined)	[-180°, 180°]	[-180°, 180°]

III. BACKGROUND

General Multiobjective Optimization Problem (MOP) is defined as Min (or Max) $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_M(\mathbf{x}))$ subject to $g_i(\mathbf{x}) \leq 0$, i = $\{1, ..., \text{G}\}$, and $h_j(\mathbf{x}) = 0$, j = $\{1, ..., \text{H}\}$ $\mathbf{x} \in \Omega$. A solution minimizes (or maximizes) the components of the objective vector $\mathbf{f}(\mathbf{x})$ where $\mathbf{x}$ is a n-dimensional decision variable vector $\mathbf{x}$ = (x_1, ..., x_n) $\in \Omega$.

In this work the terms E_b and E_{nb} (Eq. 1 and 2) represent the two objective functions to be separately minimized in the evolutionary process. The decision variables are the dihedral angles for the protein being evolved. To solve the PSP problem we use a multiobjective decomposition approach and adaptive differential evolution. The algorithms are based on the original MOEA/D and its DE version (MOEA/D-DE [16]–[18]). The adaptive part of differential evolution adopts the Probability Matching (PM) method [19]–[21] to update the probability of applying each differential evolution mutation operator.

MOEA/D is based on conventional aggregation approaches [22], it decomposes a MOP into a number of single objective optimization subproblems. The objective of each subproblem is a linear (or nonlinear) weighted aggregation of all individual objectives in the MOP. Neighborhood relations among these subproblems depend on distances among their aggregation weight vectors. Each subproblem is simultaneously optimized using mainly information from its neighboring subproblems.

DE is a stochastic, population-based search strategy developed by Storn and Price [9]. DE has three control parameters:

N, F and CR. N is the population size. The scaling factor parameter (F) is used to scale the difference between vectors, which is further added to a target vector $\hat{\mathbf{x}}$. This process is called *mutation* in DE. The vector resulting from mutation, named *trial vector*, is combined with a parent vector $\mathbf{x}^p$ in the crossover operation, according to the Crossover Rate parameter (CR). Finally, the offspring is compared with its parent vector to decide (based on their fitness) who will "survive" to the next generation.

There are some variations to the basic DE, they differ specially in the way the target vector is selected (x), the number of difference vectors used (y), and the way that the crossover point is determined (z). The notation adopted to characterize the variations is DE/x/y/z. In this paper we used DE/$rand$/1/bin, DE/$rand$/2/bin and DE/nonlinear variations. The last will be described in Section IV, while the first two variants are classical ones. For them a random individual ($rand$) is selected for target vector $\hat{\mathbf{x}}$, there is one, $y = 1$, (or there are two, $y = 2$) pair(s) of solutions which is(are) randomly chosen to calculate the differential mutation and the binomial crossover ($z = bin$) is used (the probability of choosing a component from the parent vector or the trial vector is given by a binomial distribution) [9].

The classical DE fixes all the control parameters during optimization. However, setting suitable values may lead to high computational costs due to time-consuming trial-and-error processes and may be inefficient in cases where ideal parameters' values vary during the evolutionary process. In the literature, there are some algorithms with adaptive control parameters: SaDE [23], [24], JADE [25], EPSDE [26], AdapSS [19], [27], Adap-MODE [28], ENS-MOEA/D [29], among others.

Some approaches utilize a pool of strategies. Suppose we have $S > 1$ mutation strategies in the pool $\varsigma = \{s_1, ..., s_S\}$ and the probability vector $\mathbf{p}(g) = (p_1(g), ..., p_S(g))$, where $p_s(g)$ is the the probability of using the s^{th} strategy at generation g, $\sum_{s=1}^{S} p_s(g) = 1$; $\forall g : p_{min} \leq p_s(g) \leq 1$, where p_{min} is the minimal probability of each strategy, used to ensure that no mutation strategy gets lost [20]. There is a lot of methods to adaptively update $p_s(g)$, generally based on the *empirical estimate quality* $q_s(g)$. Most of time $q_s(g)$ is updated by the rewards received, i.e., how beneficial is the use of s^{th} strategy. Denote $r_s(g)$ as the reward that the s^{th} strategy receives at generation g. Then $q_s(g)$ can be updated as follows [20]:

$$q_s(g+1) = q_s(g) + \alpha * [r_s(g) - q_s(g)] \tag{3}$$

where $\alpha \in (0, 1]$ is the adaptation rate. Based on this estimate measure, PM method uses this information ($q_s(g+1)$) to update $p_s(g+1)$ as follow [19]–[21]:

$$p_s(g+1) = p_{min} + (1 - S * p_{min}) * \frac{q_s(g+1)}{\sum_{s=1}^{S} q_s(g+1)} \tag{4}$$

Clearly, $\sum_{s=1}^{S} p_s(g+1) = 1$. From Eq 3 and 4 we see that an important issue is the received *reward*. In this paper we consider the extreme absolute reward [19], [30], defined as

$$r_s(g) = \eta_s^*(g) = \max\ [\eta_s^u(g)]_{u=1,...,|R_s|}, \tag{5}$$

where $R_s(g) = \{\eta_s^1(g)...\eta_s^{|R_s|}(g)\}$ is the set of all relative fitness improvements with cardinality $|R_s|$, and $\eta_s^u(g)$ is the u^{th} relative fitness improvement achieved by the s^{th} strategy when it is used, at generation g, to modify an individual in the population:

$$\eta_s^u(g) = \begin{cases} f_{best}/f_c(g) * \mid f_p(g) - f_c(g) \mid, & \text{if } f_c(g) > f_p(g) \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

f_{best} is the fitness of the best-so-far solution in the population, $f_p(g)$ and $f_c(g)$ are the fitness at generation g of the parent vector and its offspring (or child), respectively.

IV. The ADEMO/D Algorithm

ADEMO/D is inspired by traditional approaches for multiobjective problems, MOEA/D and MOEA/D-DE [16]–[18]. It also works under an adaptation method for selecting DE mutation strategies from a pool of strategies.

In this paper we use a slight modification of the method proposed in [11]. In the preliminary study [11] we used SaDE [23], [24] and MOEA/D [16]. In this new version, ADEMO/D is based on Probability Matching [19], [27] and MOEA/D [16]. One important advantage of using probability matching mechanism is the use of relative improvement obtained by each new solution. SaDE mechanism only considers the success rate, i.e., if the new individual is better than the previous one. With probability matching it is possible to better reward strategies that produce higher improvements. Now we also test ADEMO/D in a real problem - the Protein Structure Prediction.

ADEMO/D includes three strategies s_1, s_2, and s_3 into the candidate pool (ς): s_1: "DE/rand/1/bin"; s_2: "DE/rand/2/bin"; and s_3: "DE/nonlinear". The strategies "DE/rand/1/bin" and "DE/rand/2/bin" randomly select the target vector, and the pair(s) of individuals that influence the mutation procedure (see Section III for more details). The third mutation strategy, called in this work "DE/nonlinear", was proposed in [31] for MOEA/D framework, and disregards the values of CR and F. It is a hybrid operator based on polynomials, where each polynomial represents the offspring which takes the form $\boldsymbol{\rho}(w) = w^2\mathbf{c}_a + w\mathbf{c}_b + \mathbf{c}_c$, where w is generated based on an interpolation probability, P_{inter} [31]. Assuming that $rand \in U[0, 1]$ (i.e., $rand$ is a value between 0 and 1 randomly generated by a uniform distribution) we have $w \in U[0, 2]$, if $rand \leq P_{inter}$ and $w \in U[2, 3]$ otherwise. Individuals $\mathbf{c}_a$, $\mathbf{c}_b$ e $\mathbf{c}_c$ are defined as $\mathbf{c}_a = (\mathbf{x}_c - 2\mathbf{x}_b + \mathbf{x}_c)/2$, $\mathbf{c}_b = (4\mathbf{x}_b - 3\mathbf{x}_c - \mathbf{x}_a)/2$, $\mathbf{c}_c = \mathbf{x}_c$, where $\mathbf{x}_c$, $\mathbf{x}_b$ and $\mathbf{x}_a$ are individuals randomly chosen from the current population, in accordance with the $scope$ (variable defined during the execution of ADEMO/D, steps 12 and 14 of Algorithm 1).

The choice of "DE/rand/1/bin" is based on its usual slow convergence speed and good exploration capability. Therefore, it seems more suitable for solving multimodal problems than strategies relying on the best solution found so far [24]. According to [24] and [32], two-difference-vectors-based strategies may provide better perturbations than one-difference-vector-based strategies. Therefore, "DE/rand/2/bin" was also included. "DE/nonlinear" is a new hybrid mutation operator that includes a non-linear part for the DE mutation operator. This operator was tested with MOEA/D in [31] providing a robust performance with better results for many test problems.

Algorithm 1 details ADEMO/D in the MOEA/D platform. After initialization steps (steps 1 to 7) the algorithm enters its main loop (steps 8 to 37). Initially, all the strategies in the candidate pool have the same probability of being chosen. Step 10 selects the sth strategy used to generate a new individual. The selection is performed at current generation g based on the probability measure $p_s(g)$ updated in step 35.

Algorithm 1 Pseudocode of ADEMO/D

```
 1: Generate $N$ weight vectors $\boldsymbol{\lambda}^i = (\lambda_1^i, \lambda_2^i, ....\lambda_M^i), i = 1, .., N$
 2: For $i = 1, \cdots, N$, define the set of indexes $B^i = \{i_1, \cdots, i_C\}$
    where $\{\boldsymbol{\lambda}^{i_1}, .., \boldsymbol{\lambda}^{i_C}\}$ are the $C$ closest weight vectors to $\boldsymbol{\lambda}^i$
 3: Generate an initial population $P^0 = \{\mathbf{x}^1, \cdots, \mathbf{x}^N\}$, $\mathbf{x}^i =$
    $(x_1^i, x_2^i, ....x_n^i)$
 4: Evaluate each individual in $P^0$ and associate $\mathbf{x}^i$ with $\boldsymbol{\lambda}^i$
 5: Initialize $\mathbf{z}^* = (z_1^*, \cdots, z_M^*)$ by setting $z_j^* = min_{1 \leq i \leq N} f_j(\mathbf{x}^i)$
 6: $g = 1$
 7: For all strategies $s = 1, ..., S$, set $q_s(g) = 0$ and $p_s(g) = 1/S$
 8: repeat
 9:     for each parent vector $\mathbf{x}^i$, i=1,..,N do
10:         Select strategy $s$ from the pool according to $p_s(g)$
11:         if $rand < \delta$ then // ($rand$ in U[0,1])
12:             $scope = B^i$
13:         else
14:             $scope = \{1, \cdots, N\}$
15:         end if
16:         Generate a new solution $\mathbf{y}$ by $s$ (repair it if necessary)
17:         Apply polyn mutat to produce $\mathbf{y}'$ (repair it if necessary)
18:         Update $\mathbf{z}^*$, $z_j^* = min(z_j^*, f_j(\mathbf{y}'))$
19:         for each subproblem k do
20:             With $k$ randomly selected in $scope$
21:             if $g^{te}(\mathbf{y}' \mid \boldsymbol{\lambda}^k, \mathbf{z}^*) < g^{te}(\mathbf{x}^k \mid \boldsymbol{\lambda}^k, \mathbf{z}^*)$ then
22:                 if a new replacement may occur then
23:                     Replace $\mathbf{x}^k$ by $\mathbf{y}'$ and increment $n_r$
24:                 end if
25:             end if
26:         end for
27:         if $g^{te}(\mathbf{y}' \mid \boldsymbol{\lambda}^k, \mathbf{z}^*) < g^{te}(\mathbf{x}^i \mid \boldsymbol{\lambda}^k, \mathbf{z}^*)$ then
28:             Calculate the improvement $\eta_s^u(g)$ and group in $R_s$
29:         else
30:             $\eta_s^u(g)$ = 0
31:         end if
32:     end for
33:     Calculate the reward $r_s(g)$ for each strategy
34:     Update the quality $q_s(g)$ for each strategy
35:     Update the probability $p_s(g)$ for each strategy (PM method)
36:     $g = g + 1$;
37: until $g$ >MAX-EV
```

DE operators (mutation and crossover) are applied considering individuals randomly selected from *scope* that can swap from the neighborhood to the entire population (and vice-versa) along the evolutionary process of ADEMO/D. *scope* is composed by the indexes of chromosomes from either the neighborhood B^i (with probability δ) or from the entire population (with probability $1 - \delta$).

Based on the chosen strategy (s_1, s_2, or s_3), DE operators generate in step 16 a modified chromosome $\mathbf{y}$. The polynomial mutation in step 17 generates $\mathbf{y}' = (y'_1, ..., y'_n)$ from $\mathbf{y}$ in the following way [33]: $y'_d = y_d + \sigma_d.(y_d^{(U_p)} - y_d^{(L_w)})$, with probability p_m and $y'_d = y_d$ with probability $1 - p_m$, with $\sigma_d = (2\ .\ rand)^{\frac{1}{\tau+1}} - 1$, if $rand < 0.5$ and $\sigma_d = 1 - (2 - 2\ .\ rand)^{\frac{1}{\tau+1}}$, otherwise, where $rand \in$ U[0,1]. The distribution index τ and the mutation rate p_m are two DE parameters. $y_d^{(L_w)}$ and $y_d^{(U_p)}$ are the lower and upper bounds of the d^{th} decision variable, respectively. If an element of $\mathbf{y}$ or $\mathbf{y}'$ is out of the boundary of predetermined values, this element is repaired. After the evaluation process, if the new chromosome $\mathbf{y}'$ has an objective value better than the value stored in the empirical ideal point, $\mathbf{z}^*$ is updated with this value (step 18).

The next steps involve the population update process which is based on the comparison of fitness of individuals measured by the *Tchebycheff function*. Subproblems have the form:

$$\text{Min } g^{te}(\mathbf{x} \mid \boldsymbol{\lambda}, \mathbf{z}^*) = \max_{1 \leq j \leq M} \{\lambda_j \mid f_j(\mathbf{x}) - z_j^* \mid\} \qquad (7)$$
$$\text{subject to } \mathbf{x} \in \Omega$$

where g^{te} is the Tchebycheff function to a minimization problem, $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_M(\mathbf{x}))$ is the multiobjective function to be minimized, and $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_M)$ is the weight vector considered, which is different for each subproblem. Accordingly to *scope* (steps 12 or 14), the neighborhood or the entire population is updated. To avoid the proliferation of $\mathbf{y}'$ to a great part of the population, a maximum number of updates (NR) is used. The update is as follows: if a new replacement may occur, (i.e. while $n_r < NR$ and there are unselected indexes in *scope*), a random index (k) from *scope* is chosen. If $\mathbf{y}'$ has a better Tchebycheff [16] value than $\mathbf{x}^k$ (both using the k^{th} weight vector - $\boldsymbol{\lambda}^k$) then $\mathbf{y}'$ replaces $\mathbf{x}^k$ and n_r is incremented.

For PM method, if $\mathbf{y}'$ has a better Tchebycheff value than its parent ($\mathbf{x}^i$), the relative fitness improvement is calculated (step 28) using Eq. 6. Next, the reward value for each strategy is calculated (Eq. 5) and the quality measure is updated (Eq. 3). The probabilities of applying the different strategies are subsequently updated (Eq. 4) to be used in generation $g + 1$.

At the end of the evolutionary process, when the maximum number of evaluations (MAX-EV) is reached, ADEMO/D decision maker receives all the non-dominated solutions. In this work, the decision maker choose the final solution based on the sum of bonded and non-bonded terms (Eq. 1 and 2) of potential energy. The individual (conformation) with the lowest potential energy is chosen by the decision maker as the best conformation of the execution (run).

V. Experiments and Results

This section reports the results obtained for 10 independent runs (with different seeds for each run) of ADEMO/D algorithm while solving the PSP problem. Table III illustrates the parameter values used for all experimental tests. N, C and NR values were experimentally selected after some tests performed with ADEMO/D. The remaining parameters were set as the default values suggested by their respective frameworks.

ADEMO/D is applied to predict conformation of *Met-Enkephalin* peptide (1PLW) and *Disulphide-stabilized mini protein A domain* (1ZDD) from PDB (Protein Data Bank) [34]. *Met-Enkephalin* is a polypeptide with 5 amino acids used as classical test for algorithms designed for PSP problem. 1ZDD is a two-helix peptide of 34 residues.

Root Mean Square Deviation (RMSD) metric is used to assess how similar are the predicted conformation and the

TABLE III. Parameters used for the tests.

	Values	Description
DE Parameters		
N	60	Population size.
CR	1.0	Crossover rate.
F	0.5	Scaling factor.
P_{inter}	0.75	Interpolation probability.
p_m	$1/30$	Polynomial mutation probability.
τ	20	Distribution index of polynomial mutation.
MAX-EV	150,000	Maximum number of evaluations.
α	0.3	Adaptation rate (PM method).
MOEA/D Parameters		
C	20	Number of weight vectors in the neighborhood.
NR	2	Max number of solutions replaced by an offspring.
δ	0.9	Prob that parent solutions are selected from B^i.

native structure [8]:

$$RMSD(a,b) = \sqrt{\frac{\sum_{i=1}^{n} \mid r_{ai} - r_{bi} \mid^2}{n}} \qquad (8)$$

where r_{ai} and r_{bi} are the positions of atom i in structures a and b, respectively.

Figures 1 and 2 show the Pareto front for Met-Enkephalin peptide and for 1ZDD protein, respectively, both obtained after the stop condition in the best run is reached (i.e., 150,000 evaluations).

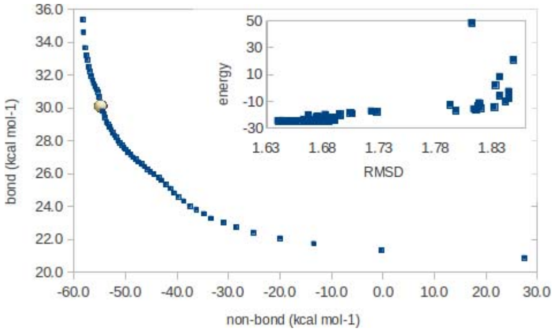

Fig. 1. Pareto front for *Met-Enkephalin* for 150,000 evaluations.

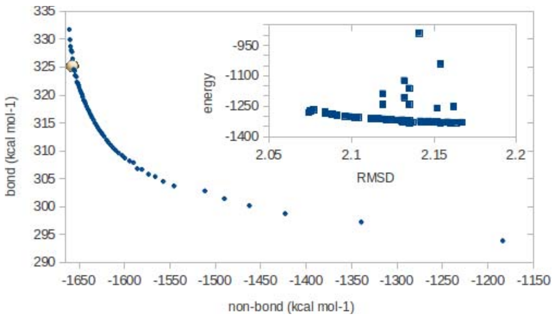

Fig. 2. Pareto front for 1ZDD for 150,000 evaluations.

It can be observed that the points in the Pareto front are well spaced except in the far right side of the curve (lowest bonded energy), maybe due to an optimizer deficiency or a discontinuity of the energy function. Figures 1 and 2 also show the plot of the energy versus RMSD$_{C_\alpha}$ for a set with 60 conformations generated by the algorithm. Relative to 1ZDD protein, the correlation between these two values indicates that the energy minimization also provides the minimization of RMSD$_{C_\alpha}$ values. We also note that better RMSD values were found in the evolutionary process, but the decision maker did not choose their conformations (see the highlighted points in Figures 1 and 2).

Figures 3 and 4 show the comparison between the predicted conformation chosen by ADEMO/D decision maker (the highlighted points in Figure 1 and 2, respectively) and protein native conformations.

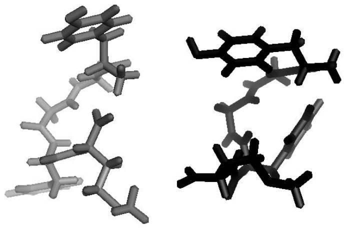

Fig. 3. Comparison between predicted conformation (left) and 1PLW conformation for Met-enkephalin peptide. Figure generated by PyMOL.

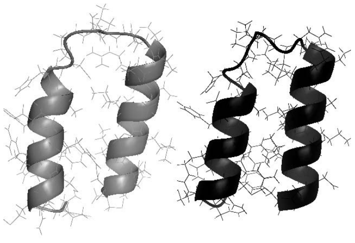

Fig. 4. Comparison between predicted conformation (left) and 1ZDD protein. Figure generated by PyMOL.

Table IV shows the results achieved by ADEMO/D when compared with those described in [5] and [35] for *Met-enkephalin* peptide and 1ZDD protein.

TABLE IV. Results for *Met-Enkephalin* peptide and 1ZDD protein.

Peptide/Protein	Algorithm	Energy (kcal mol^{-1})	RMSD$_{C_\alpha}$ ($\mathring{A}$)	RMSD ($\mathring{A}$)
Met-Enkephalin	ADEMO/D	**-24.6822**	**1.681**	**2.866**
	I-PAES [5]	-20.56	1.740	3.605
1ZDD	ADEMO/D	**-1331.3061**	**2.164**	3.280
	I-PAES [5]	-1052.09	2.27	-
	GA [35]	-983.27	3.92	-

For *Met-enkephalin* and 1ZDD, ADEMO/D proves to be a good optimizer considering the potential energy values, associated with good RMSD values.

VI. Conclusions

This paper presented the test of a multiobjective optimization method based on adaptive differential evolution and multiobjective decomposition (ADEMO/D) for the protein structure prediction problem. We used the probability matching method associated with the extreme absolute reward technique to control the adaptive selection of the best mutation strategy among those contained in the mutation strategies pool. PSP was modeled as a multiobjective optimization problem considering bonded and non-bonded contributions, which represent the two objective functions to be separately minimized in the evolutionary process. The main contribution of this paper was the evaluation of an adaptive multiobjective differential evolution for PSP problem, considering *ab initio* approach and off-lattice model. The results obtained in terms of potential energy values and their respective RMSD values, pointed ADEMO/D as a good optimizer in comparison with the literature as it found better values for all these criteria. As future work we intend to expand ADEMO/D to deal with larger proteins and investigate alternative methods for choosing the best protein conformation, i.e., alternative decision makers.

Acknowledgment

The authors acknowledge Fundação Araucária project 400/09-10705 and Myriam Delgado thanks CNPq grant 311605/2011-7 for the partial financial support.

References

[1] F. E. Cohen and J. W. Kelly, "Therapeutic approaches to protein-misfolding diseases," *Nature*, vol. 426, pp. 905–909, December 2003.

[2] B. Wang and X. Zhang, "Evolutionary computation applications in current bioinformatics," in *New Achievements in Evolutionary Computation*, P. Korosec, Ed., no. February. InTech, 2010, pp. 173–180.

[3] Y. Xu, D. Xu, and J. Liang, Eds., *Computational Methods for Protein Structure Prediction and Modeling: Volume 1: Basic Characterization*. Springer, 2006.

[4] W. Ding, D. Dai, J. Xie, H. Zhang, W. Zhang, and H. Xie, "Prt-hmm: A novel hidden markov model for protein secondary structure prediction," in *Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on*, 2012, pp. 207–212.

[5] V. Cutello, G. Narzisi, and G. Nicosia, "A multi-objective evolutionary approach to the protein structure prediction problem." *J R Soc Interface*, vol. 3, no. 6, pp. 139–51, 2006.

[6] J. Handl, S. C. Lovell, and J. Knowles, "Investigations into the effect of multiobjectivization in protein structure prediction," in *Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 702–711.

[7] D. Becerra, A. Sandoval, D. Restrepo-Montoya, and L. F. Niño, "A parallel multi-objective ab initio approach for protein structure prediction," in *BIBM*, T. Park, S. K.-W. Tsui, L. Chen, M. K. Ng, L. Wong, and X. Hu, Eds. IEEE Computer Society, 2010, pp. 137–141.

[8] A. Tramontano, *Protein Structure Prediction: Concepts and Applications*. John Wiley and Sons, 2006.

[9] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," vol. 11, no. 4, pp. 341–359, 1997.

[10] J. Moult, K. Fidelis, A. Kryshtafovych, and A. Tramontano, "Critical assessment of methods of protein structure prediction (CASP)round IX," *Proteins*, vol. 79, no. S10, pp. 1–5, 2011. [Online]. Available: http://dx.doi.org/10.1002/prot.23200

[11] S. Venske, R. Goncalves, and M. Delgado, "Ademo/d: Adaptive differential evolution for multiobjective problems," in *Neural Networks (SBRN), 2012 Brazilian Symposium on*, 2012, pp. 226–231.

[12] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," *Journal of The American Chemical Society*, vol. 117, pp. 5179–5197, 1995.

[13] M. Mijajlovic, M. J. Biggs, and D. P. Djurdjevic, "On potential energy models for ea-based ab initio protein structure prediction," *Evol. Comput.*, vol. 18, pp. 255–275, June 2010. [Online]. Available: http://dx.doi.org/10.1162/evco.2010.18.2.18204

[14] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, Dec. 1983. [Online]. Available: http://dx.doi.org/10.1002/bip.360221211

[15] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi, "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles," *Proteins: Structure, Function, and Bioinformatics*, vol. 47, pp. 228–235, 2002.

[16] Q. Zhang and H. Li, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," vol. 11, no. 6, pp. 712–731, 2007.

[17] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," vol. 13, no. 2, pp. 284–302, 2009.

[18] W. Huang and H. Li, "On the differential evolution schemes in MOEA/D," in *International Conference on Natural Computation*, 2010, pp. 2788–2792.

[19] W. Gong, lvaro Fialho, Z. Cai, and H. Li, "Adaptive strategy selection in differential evolution for numerical optimization: An empirical study," vol. 181, no. 24, pp. 5364 – 5386, 2011.

[20] D. Thierens, "An adaptive pursuit strategy for allocating operator probabilities," in *Conference on Genetic and Evolutionary Computation*, 2005, pp. 1539–1546.

[21] D. E. Goldberg, "Probability matching, the magnitude of reinforcement, and classifier system bidding," *Mach. Learn.*, vol. 5, pp. 407–425, October 1990.

[22] M. Ehrgott, "A discussion of scalarization techniques for multiple objective integer programming," vol. 147, pp. 343–360, 2006.

[23] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *IEEE Congress on Evolutionary Computation*, 2005, pp. 1785–1791.

[24] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," vol. 13, pp. 398–417, April 2009.

[25] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," vol. 13, pp. 945–958, 2009.

[26] R. Mallipeddi, P. Suganthan, Q. Pan, and M. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," vol. 11, no. 2, pp. 1679–1696, 2011.

[27] W. Gong, A. Fialho, and Z. Cai, "Adaptive strategy selection in differential evolution," in *Conference on Genetic and Evolutionary Computation*, 2010, pp. 409–416.

[28] K. Li, A. Fialho, and S. Kwong, "Multi-objective differential evolution with adaptive control of parameters and operators." in *International Conference on Learning and Intelligent Optimization*, 2011, pp. 473–487.

[29] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," vol. 16, no. 3, pp. 442–446, 2012.

[30] Y. S. Ong and A. J. Keane, "Meta-lamarckian learning in memetic algorithms," vol. 8, no. 2, pp. 99–110, Apr. 2004.

[31] K. Sindhya, S. Ruuska, T. Haanp, and K. Miettinen, "A new hybrid mutation operator for multiobjective optimization with differential evolution," vol. 15, no. 10, pp. 2041–2055, 2011.

[32] R. Storn, "On the usage of differential evolution for function optimization," in *in NAFIPS'96*. IEEE, 1996, pp. 519–523.

[33] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, ser. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001.

[34] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Res*, vol. 28, pp. 235–242, 2000.

[35] M. Dorn, L. S. Buriol, and L. C. Lamb, "A hybrid genetic algorithm for the 3-d protein structure prediction problem using a path-relinking strategy," in *IEEE Congress on Evolutionary Computation*. IEEE, 2011, pp. 2709–2716.