Swarm and Evolutionary Computation 71 (2022) 101062

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Preface

Niching methods integrated with a differential evolution memetic algorithm for protein structure prediction

Daniel Varela [a,b], José Santos [a,*]

[a] University of A Coruña, CITIC (Centre for Information and Communications Technology Research), Department of Computer Science and Information Technologies, Campus de Elviña s/n, A Coruña 15071, Spain
[b] Department of Biochemistry and Structural Biology, University of Lund, Lund, Sweden

ARTICLE INFO

Keywords:
Protein structure prediction
Niching methods
Differential evolution

ABSTRACT

A memetic version between an evolutionary algorithm (differential evolution) and the local search provided by protein fragment replacements was defined for protein structure prediction. In this problem, it is intended to find the global minimum in a high-dimensional energy landscape to discover the native structure of the protein. This problem presents a multimodal energy landscape which can additionally present deceptiveness when searching for the protein structure with minimum energy. One strategy is to try to obtain a diverse set of optimized and different protein conformations, which can be located in different local minima of the energy landscape. For this purpose, different niching methods (crowding, fitness sharing and speciation) were integrated into the memetic algorithm. The integration of niching makes it possible to obtain in a straightforward way a diverse set of optimized and structurally different protein conformations. Compared to previous studies, as well as to the widely used Rosetta protein structure prediction method, the potential solutions offered here present a diverse set of folds with different distances (RMSD) from the real native conformation, with wide RMSD distributions, and obtaining conformations closer to the native structure (in RMSD values) in some proteins.

1. Introduction and previous work

Since the biological function of proteins is related to their three-dimensional structure, knowledge of the structure of proteins can provide information about their functional role, and for this reason there is a vast amount of research on laboratory and computational methods for determining the native state of proteins. However, traditional laboratory methods used for determining this native folded structure, like X-ray crystallography and NMR spectroscopy (which require specific protocols in each case), are expensive and time-consuming. Additionally, there is an increasing gap between the number of known protein sequences (the result of many modern, large-scale DNA sequencing projects) and proteins with known structures. Different computational methods have been defined to reduce this "sequence/structure gap". There are computational methods that rely on knowing the three-dimensional structure of the homologous proteins of the target protein (homology modeling) and on finding the best fit of the target protein to some fold in a library of structures (protein threading) [1]. The problem is that these methods require the existence of protein template structures, such as proteins with similar sequences and known structures for homology modeling, which is not the case for many novel protein sequences.

Hence, there has been considerable research using computational "ab initio" or "de novo" methods, which only use the protein primary structure (protein sequence of amino acids) as input information, with the aim of accurately determining the final native state of the protein, which is a formidable challenge in computational biology. It is assumed that the protein's amino acid sequence is the only relevant information that determines the final folded structure, as shown by Anfinsen [2] (Anfinsen's dogma). In this computational Protein Structure Prediction (PSP), it is also assumed that the protein's native state corresponds to the one with the lowest Gibbs free energy [2]. Consequently, the problem of the folded structure prediction becomes a search or optimization problem in an energy landscape associated with possible protein conformations, since the goal is to find the global minimum in the energy landscape, minimum that is assumed to correspond to the native structure of the protein. The energy landscape takes into account the simplifications and constraints of the lattice (the protein components are located in the sites of a lattice model) or off-lattice model (amino acids and their atoms can be located without that restriction) used to represent a protein conformation. This energy minimization approach in PSP is different from the alternative deep learning-based prediction of the resolved protein structure (usually the crystallized structure), in the latter case with its

* Corresponding author.
E-mail addresses: daniel.varela@udc.es (D. Varela), jose.santos@udc.es (J. Santos).

https://doi.org/10.1016/j.swevo.2022.101062
Available online 24 March 2022
2210-6502/© 2022 Elsevier B.V. All rights reserved.

dependence on multiple sequence alignment information from the input sequence [3].

In such a search, the ability to pursue global exploration of evolutionary methods (and population-based methods in general) reduces the chance of the search becoming stuck at local minima [4–7]. The energy landscapes associated with protein structure are full of local minima and are usually regarded as "funnel-like" [6]. As Zhang et al. [8] state, "Protein structure prediction involves an extremely expensive to evaluate energy model which has thousands of degrees of freedom, and a highly degenerate energy hyperspace owing to its massive local minima and large regions of unfeasible conformations". There has been extensive research on the use of evolutionary methods and other bio-inspired algorithms in protein structure prediction using lattice models for protein representation [5,7,9–15], including hybrid or memetic solutions [16] that integrate the combination with a local search or a procedure that refines unfeasible protein conformations to legal ones [17–21]. However, there has been limited research on the use of evolutionary methods with off-lattice protein representation models [22–24] and, in particular, the widely used model of the Rosetta software environment for computational modeling and analysis of protein structures [25,26].

Rosetta [25] is one of the most successful software environments for protein design, widely used for PSP and validated during CASP (Critical Assessment of Structure Prediction) competitions [27] as one of the leading approaches. For determining the native structure, the Rosetta ab initio protocol [25,26] basically uses a local search technique with its low-resolution protein representation, based on a Metropolis Monte Carlo procedure with the use of a fragment replacement technique (explained in Section 2.1) [26,28]. The procedure is repeated thousands of times due to the stochasticity of the process, and some of the final conformations of this phase ("decoys") can be used in a second "ab initio relax" procedure that uses the Rosetta's full atomic model. In the Rosetta system there are inaccuracies in its energy formulation, especially given its "knowledge-based" nature (Section 2.1), in which the lowest energy structures are not necessarily the most native-like [29]. That is, the energy function is not accurate enough to differentiate between the actual native structures of proteins and the conformations structurally close to the native state. The Rosetta system employs the decoy set, obtained in its first search with the low-resolution model, to increase the probability of obtaining solutions close to the native one.

Different studies used evolutionary algorithms with the low-resolution model of Rosetta and also with the fragment replacement technique [22,30]. Moreover, given the deceptiveness of the Rosetta energy landscape, in the sense that the global minimum is not necessarily the one that corresponds to the native structure, previous work has focused on increasing the diversity of the conformations maintained in the genetic population, with the aim of addressing the problem by obtaining a diverse set of native-like structures.

Thus, Garza-Fabre et al. [31] defined a multistage memetic algorithm which incorporates Rosetta fragment replacements as a local search routine. Their analysis showed that specialized genetic operators (adapted recombination and crossover), applied only in residues located at predicted loop regions, increase the exploration in such loop regions in order to discover different protein folds. Moreover, the authors used a stochastic ranking-based survival selection with two selecting criteria (energy and diversity), where the latter aims to improve the exploration and preservation of different protein folds throughout the evolutionary process. In a later study, Kandathil et al. [32] developed two sampling protocols (bilevel optimization and iterated local search) to improve the exploration capability of protein conformations. Their bilevel protocol allows "Perturbation steps" in (predicted) loop regions, while "LocalSearch" steps are applied in the rest of the protein. Comparisons with the Rosetta ab initio method indicate that their protocols more frequently generate native-like predictions for many targets than in the case of Rosetta.

Other examples focused on increasing the diversity of optimized conformations include the selection strategy of Zhang et al. [8], in which solutions with worse energy but more "reasonable structure" (in the authors' words) can be selected for the next generation of their evolutionary algorithm. Their results showed that their strategy allowed sampling near-native conformations more effectively than Rosetta in most of the benchmark proteins used. In Simoncini et al. [33], a cluster-based scheme for protein model selection in a population-based metaheuristic also tends to explore more diverse low-energy protein folds. Finally, in the memetic algorithm approach of Corrêa et al. [34], the authors enforced structural diversity by maintaining subpopulations with different packing degrees of the protein structures, these degrees being measured with the protein radius of gyration.

Nevertheless, there is an unexplored possibility when evolutionary methods are used in protein structure prediction: the explicit use of niching methods for the simultaneous search for the best solutions in a multimodal fitness landscape, with the aim of obtaining a set of diversified conformational folds. Niching refers to the possibility of clustering the population around promising solutions in the search space. For this reason, several techniques for generating niches during the evolutionary process were developed in the evolutionary computation field, especially when dealing with a multimodal fitness landscape and when the objective is the simultaneous location of the fitness peaks or local maxima/minima (or at least the highest/lowest peaks as well as the global optimum). Consequently, our study employs niching methods with the objective of directly enforcing structural diversity in the conformations, which can be located at different peaks of the multimodal fitness landscape, this objective being the main contribution of our study.

It should be noted that the aim of the present paper is not to compare different niching methods in the application. Thus, standard and widely used niching methods (crowding, fitness sharing and speciation) were selected with the aim of enhancing structural diversity in the population of protein conformations during the evolutionary search process, concentrating the search for protein conformations on those promising areas or niches found by the evolutionary algorithm, with the objective of improving the probability of obtaining native-like solutions. The incorporation of niching has two effects: i) there is an inherent diversity in the population, since the individuals are searching in different areas of the landscape that correspond with different protein folds and, ii) given the inaccuracies of the fitness/energy landscape of knowledge-based energy formulations like the one used in Rosetta, in which the native conformation is not necessarily located in the energy minimum, the location of solutions in different promising areas can increase the probability of obtaining solutions close to the native structure. Contrary to previous studies that used niching methods in PSP considering the first effect [35,36] (avoiding premature convergence), our current study takes advantage of both effects to obtain the required diversified set of optimized folds.

Differential Evolution (DE) [37,38] was selected as the evolutionary algorithm for the search for optimized protein conformations. It should also be noted that this study is not a comparison between different evolutionary algorithms or other metaheuristics or DE versions. Standard DE was chosen as it is one of the best contrasted methods in evolutionary computation and with few tuning parameters [39,40], and also has better results in PSP with the simple HP lattice model compared to approaches based on genetic algorithms (GAs) [19,41]. Moreover, in a preliminary study [42] we combined DE [37,38] with Rosetta's fragment replacements. This hybrid version DE/fragment replacements [42] obtained better results in terms of energy, with respect to the Rosetta ab initio procedure and other solutions with evolutionary algorithms, and under the same number of fitness evaluations. However, although the hybrid version of the evolutionary algorithm has a better ability to sample the energy landscape in order to find solutions with minimized energy [42], this is not a guarantee of finding the best native-like conformations for the target protein, due to the inaccuracies of the knowledge-based energy formulation of Rosetta. Therefore, we tested the integration of a memetic version of differential evolution and the aforementioned classic niching methods, in this context of protein

structure prediction with the Rosetta atomic off-lattice model, trying to obtain a diversified set of folds. The memetic combination allows incorporating the advantage of both methods: the global search of DE with the problem-specific local search (provided by fragment replacements for local refinement of protein structures), integration that allows a more efficient sampling of the energy landscape. Moreover, the incorporation of the niching methods into the memetic version aims to obtain diversified and optimized protein structures.

There has been ample research integrating niching techniques into evolutionary algorithms [43,44] and, more specifically, with DE [45–47]. In the case of crowding [48], the standard integration with DE defined by Thomsen [49] was followed in the present study (*CrowdingDE*), extending DE with the classic crowding scheme. In *CrowdingDE* each trial individual competes with its nearest member of the population in order to preserve diversity and the location of promising solutions (Section 2.3). Some preliminary results with *CrowdingDE* in PSP were published in [50,51] using a limited set of proteins. In this study an extension of the results is provided, with an ample protein dataset and with a comparison with different Rosetta-based ab initio protocols.

Fitness Sharing (FS) [52,53] was another of the methods integrated into evolutionary algorithms, and specifically into DE [49,54]. The basic idea in FS is to punish individuals that occupy the same area of the search space, rescaling the fitness of each encoded solution taking into account the number of individuals in its neighborhood (Section 2.4). The version called *SharingDE* and defined by Thomsen [49], with the direct integration of FS with DE, is the one that we followed here (explained in Section 2.4).

Finally, in speciation, each of the "species" is built around a dominating species' seed. All individuals that fall within the radius from the species seed are identified as the same species. Since DE mutation is carried out within each species, the technique has the ability to maintain high diversity and stable niches over generations [45]. The Species-based DE (*SDE*) defined by Li [55] was used as a base in our approach, explained in Section 2.5.

Therefore, taking into account all these considerations and previous work, the objectives of our study are the following: i) To test the ability of the memetic version of DE to obtain protein structures with minimal energy and to compare its search capability with the state-of-the-art Rosetta ab initio protocol (and recent variants thereof), both as energy minimization approaches in PSP. ii) To explore the incorporation of niching to address possible deceptiveness in the energy model. For this purpose, standard niching methods are integrated into the memetic algorithm, with a detailed analysis of the capabilities of this incorporation and the dependence of the results on the defining parameters of the niching methods. iii) Finally, a key aspect to explore is the ability of niching to provide an optimized, but also structurally diverse set of proteins.

The remainder of this article is structured as follows. Section 2 details the methods used: a summary of the Rosetta software environment, our approach to sampling the energy landscape based on a combination of differential evolution and Rosetta's fragment replacements, detailing the encoding of protein conformations in the genetic population, the population initialization and the different stages considered in the hybrid evolutionary algorithm. Section 2 also explains in detail the integration of the considered niching methods into the hybrid DE algorithm. Finally, in this section, the protein sequences used as benchmarks in the study are detailed. In Section 3, the experimental results with the different niching alternatives are presented, while in Section 4 a final discussion is provided together with the main conclusions.

2. Methods

2.1. Main aspects of Rosetta

Rosetta uses two off-lattice protein representations: coarse-grained and all-atom. In the low-resolution coarse-grained representation, only the main atoms of the backbone chain are represented (with their dihedral angles), whereas the side chains are described by a centroid located at their center of mass (Fig. 1). Therefore, protein conformations are defined in the dihedral space, and a protein chain has three degrees of freedom (ϕ, ψ and ω) for each amino acid residue. The all-atom representation also includes rotation *Chi* angles for side chains. All the other internal degrees of freedom are fixed to "ideal" values (e.g., all bond lengths and bond angles are fixed) [56].

Rosetta ab initio protocol [25,26], with the low-resolution protein representation, employs a search technique in which a Monte Carlo procedure decides whether the dihedral angles of small protein fragments can replace the original ones [26,28]. A structural fragment is a continuous subset of the residues of a protein. Those fragments are drawn from experimentally determined structures (a non-redundant set of proteins). The selection of those fragments is based on the sequence similarity between the fragment and the region (window of amino acids) of the target sequence where it is going to be inserted. That position for fragment insertion in the target is randomly chosen. Rosetta uses fragment regions of 3 and 9 residues long. Those fragments are extracted for each position in the target, a process that is prior to the ab initio run, and it generates a library of fragments (particular to each target protein). Typically, there are many fragments in the library for each position in the target (e.g., Rosetta's fragment libraries contain around 200 fragments per position)

The decision regarding whether the dihedral angles of a selected fragment replace the ones in the target protein is based on the Metropolis criterion [57]. This criterion always accepts the changes that improve the energy (lower values), while occasionally accepting dihedral angle changes that worsen the energy (energy increase), following a Boltzmann energy distribution for a given temperature. This procedure makes it more likely that the search of protein conformations with the fragment replacement technique can escape from local minima.

Regarding the energy model, Rosetta uses physics and knowledge-based energy terms [58]. Knowledge-based potential [59] refers to the empirical energy terms derived from the statistics of the resolved structures deposited in the Protein Data Bank [60]. The interesting property is that these knowledge-based terms require less computational time. Common physics-based energy terms [61] are associated with bond lengths, angles, torsion angles, van der Waals and electrostatic interactions.

The Rosetta energy score associated with a protein conformation is defined as a linear weighted combination of such terms that models molecular forces that act on and between all atoms in that conformation. These scoring terms are, in most cases, knowledge-based. There are energy terms such as solvation and electrostatics effects, repulsion and hydrogen bonding, as well as secondary structure scores like strand pairing and helix-strand packing. Steric overlap of backbone atoms and side-chain centroids is penalized, but favorable van der Waals interactions are modeled only by rewarding globally compact structures [26]. The Rosetta score function which takes into account all energy components, called *score3*, corresponds to the full coarse-grained energy function. Nevertheless, Rosetta changes the weight set depending on the stage of its ab initio protocol, as detailed in what follows.

It is well-known that the Rosetta's knowledge-based energy model is inaccurate since the native conformation is not necessarily located in the minimum of energy. For example, the results of Shmygelska and Levitt [29] reveal some of the deficiencies of the existing energy terms in Rosetta, including the presence of false local minima and a general flatness of the energy landscape near the native states.

To search through the conformational space, many rounds of Metropolis Monte Carlo are performed. For this purpose, the Rosetta ab initio protocol is divided into four stages. Through these stages, Rosetta uses the coarse-grained protein representation and its fragment insertion technique, together with the Metropolis criterion [57], to generate new decoys. Table 1 includes a short summary of each stage with the most important details.

Moreover, the number of fragment insertion attempts in each stage (Table 1) can be modified with the Rosetta parameter *increase_cycles*,

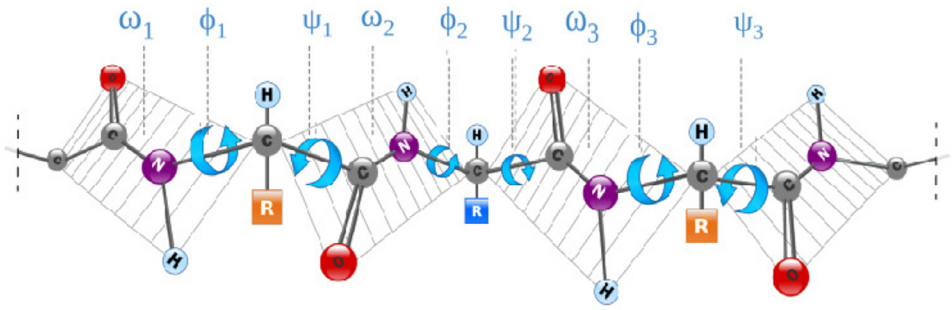

Fig. 1. Low-level coarse-grained representation of Rosetta. Only the main backbone atoms are considered, while the lateral chains are represented with a pseudo-atom. Each protein conformation is encoded with the dihedral angles of the different amino acids: ω (between atoms of the peptide bond), ϕ and ψ.

Table 1
Main aspects of Rosetta ab initio stages.

S1	Begins with a fully extended chain and inserts 9-mer fragments until all the backbone angles are modified at least once and with a maximum of 2000 cycles (fragment insertion attempts). During this stage, the energy function (called *score0*) only considers the steric-clash term to prevent overlapping between backbone atoms and side-chain centroids.
S2	Employs 9-mer fragment insertions over 2000 cycles, but uses a more complex score function, *score1*, which adds terms such as hydrophobic burial and specific pair interactions, as well as secondary structure scores.
S3	Runs 10 iterations of 2000 cycles of 9-mer fragment insertion attempts. Rosetta combines in this stage two score functions, *score2* and *score5*. These focus on compactness and secondary structure terms. A convergence check determines the structural similarity of the current conformation with respect to a reference one, regularly updated. If there is not enough structural variation after 100 fragment insertions are accepted, stage 3 ends.
S4	Employs 3-mer fragment insertions over 12,000 cycles, split into 3 iterations of 4000 cycles for each one. In this stage, *score3* (which takes into account all energy components) is used.

which multiplies the default values of cycles in the different stages. This ab initio procedure, with the four stages, is repeated thousands of times due to the stochasticity of the process. Some of the final conformations ("decoys") of this phase (where a clustering process can be used to decipher the most representative decoy set), can be subjected to a refinement, with the side chain reconstruction and all-atom energy minimization, in a second "ab initio relax" procedure that uses the full Rosetta's atomic model.

2.2. Hybrid DE using different Rosetta phases and energy functions

The same structure for the combination between an evolutionary algorithm and the Rosetta ab initio search defined by Garza-Fabre et al. [31] was followed. Those authors used a memetic algorithm, combining a genetic algorithm with the local refinements provided by fragment replacements [31].

In our case, Differential Evolution [37,38] was used as evolutionary algorithm (with the next incorporation of niching into the evolutionary process). DE is a population-based search method that creates new candidate solutions by combining existing ones, according to a simple formula of vector crossover and mutation, and then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand. The central idea of the algorithm is the use of difference vectors for generating perturbations in a population of vectors. DE needs a reduced number of parameters to define its implementation. Apart from the population size, the parameters are F or differential weight and CR or crossover probability. The weight factor F (usually in [0,2]) is applied over the vector resulting from the difference between pairs of vectors (x'_2 and x'_3 in Algorithm 2.1). CR is the probability of crossing over a given vector of the population (target vector x') and a "mutant" vector created from the weighted difference of two vectors ($x'_1 + F(x'_2 - x'_3)$). The (most usual) "binomial" crossover (specified in Algorithm 2.1) was used for defining the value of the "trial" vector (y) in each vector component or position i [39]. The index R guarantees that at least one of the vector parameters will be changed in the generation of the trial solution.

In this protein structure prediction problem, the solutions of the population encode protein conformations. Each solution is encoded with the three dihedral angles, ϕ, ψ and ω, for each amino acid. The application of forward kinematics to this angular representation obtains the spatial information of the protein conformation. DE individuals code the dihedral angles in the range [-1,1], which are decoded to the interval $[-180°, 180°]$ (degrees). The angles ϕ and ψ are evolved with DE whereas the third dihedral angle, ω, is not evolved. This last angle is only changed by fragment replacements, as detailed below, this being the case because this angle can only have two configurations of 180° or −180°.

DE was combined with the local refinement procedure provided by the Rosetta ab initio stages with their fragment replacements, defining the hybrid DE version in the application. The pseudo-code of Algorithm 2.1 summarizes the procedure of the hybrid DE version. The initialization of the population is defined with Stage 1 of Rosetta (using *score0* as energy), that is, applying fragment replacements over the initial extended conformation in order to provide a diversified set of structures. Then, the hybrid DE version employs 3 sequential stages (Stage 2, Stage 3 and Stage 4) with the same number of evolutionary generations (MAX_GEN), using in each stage the corresponding Rosetta score functions (Table 1).

In the first generation of each stage, no trial individuals are generated. In generation 1, all the individuals are refined with the Rosetta replacement procedure of the corresponding stage, that is, with the corresponding score function, number of fragment insertion attempts and fragment lengths (Section 2.1, Table 1). Each individual x is therefore refined to an individual x'. The next generations are standard DE generations since, for each target individual x', a trial individual y is defined after the DE mutation and crossover operators. The Rosetta replacements are now applied (with the corresponding score function and fragment cycles of the current Rosetta stage) only to the trial individuals y, generating refined trial individuals y'. As in standard DE, the fitness of the final trial individual y' is compared with the fitness of the corresponding target individual x', to determine which one enters the population in the next generation.

When defining the "mutant" vector, the DE scheme that chooses the base vector x'_1 randomly was used (variant $DE/rand/1/bin$, where 1 denotes the number of differences involved in the construction of the mutant vector, and bin denotes the crossover type), which provides the lowest selective pressure. Note that the fundamental idea of DE is to adapt the step length ($F(x'_2 - x'_3)$) intrinsically along the evolutionary process [40]. At the beginning of generations the step length is large,

Algorithm 2.1: Hybrid Differential Evolution and CrowdingDE

```
for each Individual ∈ Population
  do {Individual ← InitializeRandomPositions - Rosetta Stage 1()

// DE with three stages, using the score and replacement cycles of the corresponding
// Rosetta stages
for each s ∈ 2 : 4   // For each Rosetta stage 2 to 4
    do { for each Individual ∈ Population
           do { x' ← Rosetta Stage s(x)
                // Rosetta stage s is applied to all individuals in DE generation 1
         for each g ∈ 2 : MAX_GEN   // For each DE generation g
             do { for each Individual  x' ∈ Population
                    do { x'_1, x'_2, x'_3 ← GetRandomIndividual(Population)
                         // x'_1, x'_2, x'_3 must be distinct from each other and x'
                         R ← GetRandom(1, n)  // n - problem dimensionality
                         for each i ∈ 1 : n
                         // Compute individual's potentially new position
                         // y = [y_1, ..., y_n] (trial vector)
                           do { r_i ← GetRandom(0, 1)// uniformly in (0,1)
                                if ((i = R) || (r_i < CR))  // CR - crossover prob.
                                    y_i = x'_{1_i} + F(x'_{2_i} − x'_{3_i}) // F - weight factor
                                else y_i = x'_i
                         y' ← Rosetta Stage s(y)
                         // trial vector y is refined with Rosetta stage s to generate y'
                         if (f(y') ≤ f(x'))   x' = y'
                         // if y' has better fitness, replace x' with y'
                         // In CrowdingDE, the refined trial vector y' is compared
                         // (regarding fitness) with its most similar vector in the
                         // subset defined by parameter CF
```

because individuals are far away from each other. As the evolution continues, the population converges and the step length becomes smaller and smaller, providing in this way an automatic balance between exploration and exploitation in the search.

The refinement procedure can be considered as a local search since it refines the dihedral angles of a protein for obtaining a better (in energy terms) conformation. This inclusion of a sequence of consecutive backbone dihedral angles (fragment) simultaneously simplifies the process of generating physically-realistic conformations with respect to angle values sampled uniformly at random [22]. Therefore, the combination provides a more efficient search, integrating the advantages of DE as a global search method with the problem-specific local search of short conformations provided by fragment replacements. The refinement procedure is applied to the encoded conformations of the population (in generation 1 of the three stages) and to the trial individuals defined by the DE genetic operators. Since the refined individuals replace the original ones (in the case of trials it depends on whether these improve the energy of their target individuals) the hybrid DE version is a Lamarckian combination [62]. However, since the new refined genetic material only replaces the original one in the first generation, together with the stochasticity of the Monte Carlo refinement process, it allows us to overcome the possibility of a fast loss of genetic variability, a problem inherent to a Lamarckian strategy [62].

Fig. 2 (uppert part) illustrates the workflow of the optimization process with the memetic algorithm, along with an example of the fitness evolution in the evolutionary process with the three sequential stages. It shows the best fitness and the average fitness of the population across generations (bottom part). In the example, the encoded protein conformations (population of 100 individuals) evolved through 100 generations in each of the three evolutionary stages, that is, a total of 300 generations. Note that the fitness evolution presents different ranges in the transition between the evolutionary stages, since the energy/fitness corresponds to the particular score definition in the same Rosetta stage. This use of different score functions in different consecutive stages, as in Rosetta ab initio, allows the progressive refinement of conformations. The Results section describes the setup of the different parameters for the appropriate comparison between the hybrid DE version and Rosetta.

2.3. Crowding and CrowdingDE

In our application, we used the same integration employed by Thomsen [49] of the classic crowding niching method [48] of evolutionary computation with the differential evolution algorithm [37].

Crowding, introduced by De Jong [48], is one of the most widely used niching methods for dealing with multimodal optimization problems in evolutionary computation. Since the goal of niching methods is

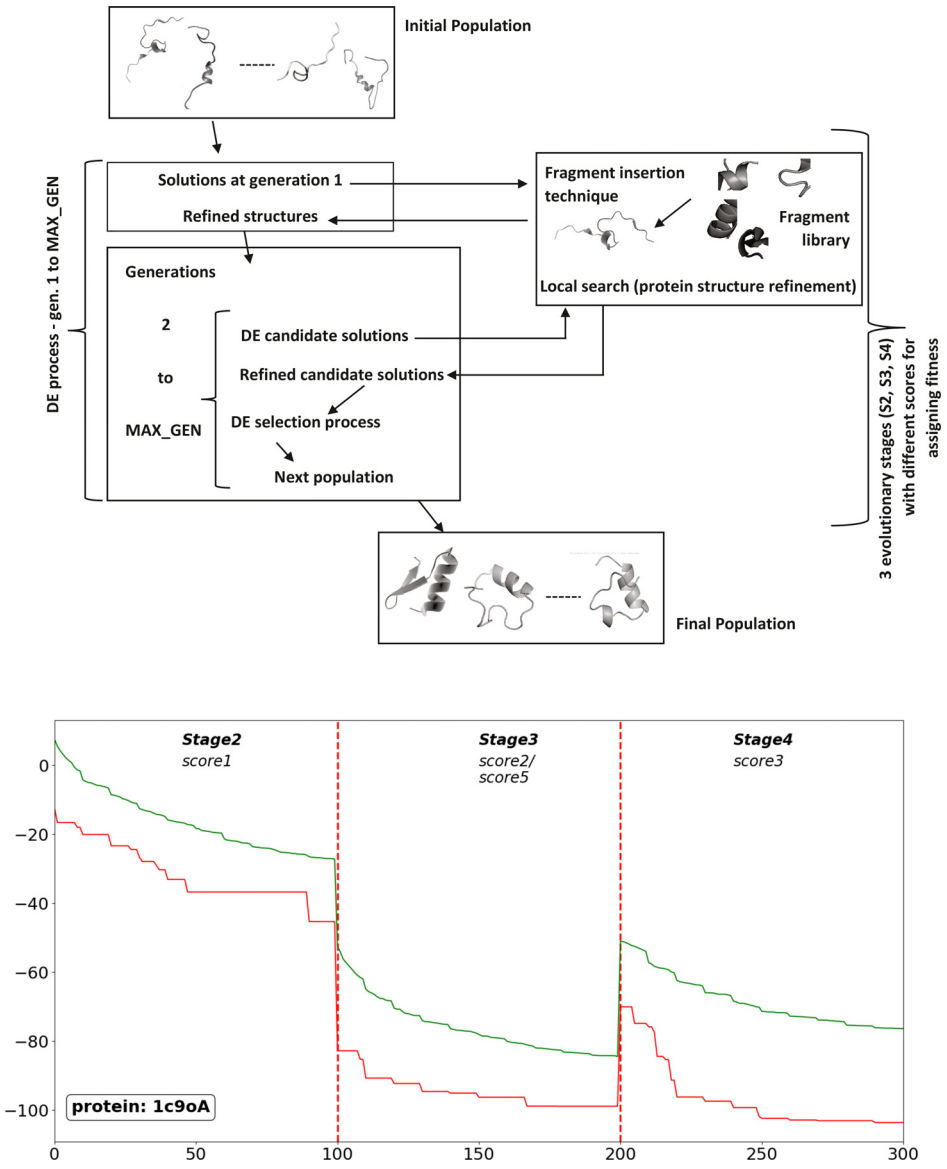

Fig. 2. Upper part: Workflow of the optimization process to obtain protein structures with minimum energy. Bottom part: Fitness evolution in the hybrid DE approach: each of the three sequential evolutionary stages corresponds to the same Rosetta stage, since the same fragment lengths and score energy functions of the corresponding Rosetta stage were used in each evolutionary phase. The graphs correspond to the evolution of the energy/fitness of the best individual (red line) and the average energy/fitness of the population (green line) in a run with protein *1c9oA* as target.

to simultaneously search within the most promising areas of a multimodal fitness landscape, the simple idea behind crowding is the preservation of genetic diversity, where an offspring individual replaces the most similar individual in the population if it has a better fitness. That (most similar) individual is selected among a randomly chosen subset of the population, where the size of the subset is determined by a parameter called Crowding Factor (CF).

A distance measure between two protein conformations must be defined for this purpose. It can be the root mean square deviation between the c_α carbons of the backbone chains or alternative measures, such as the one which we will defined below (Section 2.6).

The integration with DE (*CrowdingDE*)[1] is simple [49], extending DE (in our case hybrid DE) with the classic crowding scheme. With respect to the hybrid DE version defined in the previous section, the only modification in *CrowdingDE* is that now each refined trial individual in DE is compared to its nearest (most similar) neighbor among that subset CF of the population, to decide which one enters the next generation (see the final comment in Algorithm 2.1). If the refined trial individual ($f(y')$) is fitter, then it replaces the closest one from the subset CF. That is, Algorithm 2.1 is the same for the hybrid DE version and *CrowdingDE* except for the final comparison between the DE refined trial vectors and the corresponding target vector (x' in the case of hybrid DE, the most similar vector in the subset CF in the case of CrowdingDE). Algorithm 2.1 illustrates this aspect in the final comment of the pseudocode.

With low CF values, the offspring (in *CrowdingDE*, the refined trial individuals) can replace an individual of the population that is not very similar to the new offspring, resulting in so-called crowding replacement errors. Such replacement errors lower the population diversity and, consequently, favor premature convergence. Although there are alternatives to reduce replacement errors [45], the simplest one (used here and in [49]) is to consider CF as the whole population. In this case, the complexity is $O(N^2)$, with N being the population size, since the distance of each of the N trials to the whole population must be calculated.

[1] The code of the different niching methods integrated with the defined hybrid DE version can be downloaded from https://github.com/danielvarela/RosettaEvolution. The code was parallelized in MPI (Message Passage Interface), with a main master process that distributes the population of individuals, in equal size chunks, to different slave processes in order to evaluate the fitness of each individual.

2.4. Fitness sharing and SharingDE

Fitness sharing is also a classic technique in evolutionary computation for dividing the population into different subgroups according to the similarity of the individuals. This concept of fitness sharing was introduced by Holland [52] and expanded by Goldberg and Richarson [53]. The shared fitness for the ith individual is defined as:

$$f_{shared}(i) = \frac{f_{original}(i)}{\sum_{j=1}^{N} sh(d_{ij})} \quad (1)$$

where the sharing function is calculated as:

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right)^{\alpha} & if\ d_{ij} < \sigma_{share} \\ 0 & otherwise \end{cases} \quad (2)$$

being d_{ij} the distance between individuals i and j, σ_{share} the sharing radius, N the population size, and α a constant called the sharing level.

Finally, if the application requires the minimization of the fitness rather than its maximization, the formula in Eq. 1 is a multiplication between the two terms.

In this way, fitness sharing modifies the search landscape by reducing the payoff in densely populated regions. The main drawback of the technique is its complexity, $O(N^2)$, because of the calculation of inter-distances. On the other hand, two important properties result from the incorporation of FS into the DE algorithm: i) Fitness sharing tends to encourage searches in unexplored regions of the space and favors the formation of stable subpopulations [63]; ii) Since the optimization capability of DE depends extensively on the diversity between vectors, if the diversity of the population descends too fast, it may lead to a high possibility of obtaining a local optimal solution [54]. The incorporation of FS makes such a situation of diversity loss more difficult, since FS tends to locate individuals in different areas of the search landscape.

With the inclusion of FS in the DE functioning, the same *SharingDE* version described in [49] (Algorithm 2.2) was followed. In standard DE, each generated trial or candidate individual is compared with the target vector (Algorithm 2.1). In *SharingDE*, for a population of N individuals, N trials are generated (with the same mutation and crossover procedure as standard DE) and are added to the population, enlarging the population by a factor of two. As in *CrowdingDE, SharingDE* incorporates the hybridization with the fragment replacement procedure, since these N trials are refined with fragment insertions.

In such a doubled population, the FS technique is applied, since the fitness of the $2N$ individuals is rescaled according to Eq. 1. The enlarged population is then sorted with respect to that effective fitness. The final step is removing the worst half of the enlarged population, so that the population size is maintained constant. In this way, a trial that enters an area that is not largely populated can survive, since it has better effective fitness with respect to other individuals with better original fitness but that are located in a densely populated region. Algorithm 2.2 summarizes the approach, where the lines that imply a change with respect to the hybrid DE version without fitness sharing (Algorithm 2.1) are emphasized in bold.

Moreover, it should be noted that, because of such a fitness scaling, elitism is needed to preserve the overall best solution found, so the best solution (regarding the original fitness) replaces the worst individual in the new population in the case that the best individual was removed during the selection process in the doubled population.

Finally, regarding these niching methods, there is a reason why the local search with the fragment replacement procedure is not applied in all generations (only in generation 1 of the three stages, Algorithms 2.1 and 2.2) to refine the individuals of the population. The local search can change a fold to another one which is already present in the current population, so it can "destroy" the work of the niching method that conserves individuals in different niches (different folds). However, that application of the local search to the whole population is useful since it can refine all the individuals to better positions (in energy terms), especially poor individuals that are isolated and which can never be selected, for example, as the nearest ones in *CrowdingDE*. Therefore, the application of the local search to the individuals of the whole population is maintained, but only in the first generations of the three evolutionary stages.

2.5. Speciation and Species-based DE (SDE)

The final niching method considered is *speciation* [55,64,65]. In biology, a species is defined as a group of individuals of similar biological features capable of interbreeding among themselves, but not with individuals from a different group. It is a concept obviously interrelated with the niching concept, although, as Horn [66] states "A niche can be defined generally as a subset of resources in the environment. A species, on the other hand, can be defined as a type or class of individuals that takes advantage of a particular niche. Thus, niches are divisions of an environment, while species are divisions of the population".

The Species-based DE (*SDE*) defined by Li [55] was used as a base. *SDE* incorporates the speciation concept to handle multimodal landscapes, since it locates different optima simultaneously through an adaptive formation of species. One important aspect of *SDE* is that each species is evolved through its own DE process, which seeks to successively improve itself.

Algorithm 2.3 shows the steps for determining the species seeds. Such species are treated as subpopulations running DE (hybrid DE in our case) independently themselves, that is, applying the DE operators (regarding the generation of mutant and trial vectors) only with the individuals of the same species. However, contrary to Li's version [55], which allows a variable number of species and a variable number of associated individuals across the evolutionary process, the number of species and their subpopulation size remain fixed. The reason for this is that our version facilitates the parallelization of the code, with subpopulations (species) with the same number of individuals that can run their DE processes in parallel.

The *SDE* method depends on a radius parameter r_s, which measures the distance from the center of a species (called the seed) to its boundary. The center of the species or seed is always the fittest individual of the species. Individuals that fall within the radius from the species seed are (generally) identified as the same species, so each of the species is built around the dominating species' seed. The different steps in *SDE* can be summarized as:

1. Generate the initial population with randomly generated individuals.
2. Evaluate all individuals in the population.
3. Sort all individuals in descending order of their fitness values (i.e., from the best-fit to least-fit ones).
4. Determine the species seeds for the current population (Algorithm 2.3).
5. Each species is filled, first, with the nearest individuals (to the seed) from the whole population (except the seeds of other species) with distances lower than r_s.
6. The remaining individuals, not assigned in the previous step to a species, are associated with the (not completed) species whose seed is the closest. This implies that a species can have associated individuals with higher distances than r_s to the species seed, but this also allows an increase in exploration.
7. If an individual is very close, in energy terms, to the corresponding seed (a small threshold is used), then the individual is randomized (as in [55]).
8. For each species, a basic hybrid DE (Algorithm 2.1) is run for a given number of generations (*NUMBER_GEN*).
9. Go back to step 2, unless the termination criterion is met.

Note that, if r_s is too small (Algorithm 2.3), then the species seeds can correspond to the individuals with the best fitness, but these can be very close to each other, which is not the aim with niching. On the contrary, with large values of r_s, the seeds will correspond to progressively

Algorithm 2.2: Hybrid Differential Evolution - SharingDE

for each $Individual \in Population$
do $\{Individual \leftarrow$ InitializeRandomPositions - Rosetta Stage 1()

// DE with three stages, using score and replacement cycles of the corresponding
// Rosetta stages
for each $s \in 2:4$ // For each Rosetta stage 2 to 4
do
- **for each** $Individual \in Population$
  do $x' \leftarrow$ Rosetta Stage $s(x)$
  // Rosetta stage s is applied to all individuals in DE generation 1
- **for each** $g \in 2:GEN\_MAX$ // For each DE generation g
  do
  - **for each** $Individual\ \ x' \in Population$
    do
    - **Doubled Population** $\leftarrow$ **Add(x')**
    - $x'_1, x'_2, x'_3 \leftarrow$ GetRandomIndividual($Population$)
    - // x'_1, x'_2, x'_3 must be distinct from each other and x'
    - $R \leftarrow$ GetRandom$(1, n)$ // n - problem dimensionality
    - **for each** $i \in 1:n$
    - // Compute individual's potentially new position
    - // $y = [y_1, ..., y_n]$ (trial vector)
    - **do**
      - $r_i \leftarrow$ GetRandom$(0,1)$// uniformly in (0,1)
      - **if** $((i = R)\ ||\ (r_i < CR))$ // CR - crossover prob.
        do $y_i = x'_{1_i} + F(x'_{2_i} - x'_{3_i})$ // F - weight factor
      - **else** $y_i = x'_i$
    - $y' \leftarrow$ Rosetta Stage $s(y)$
    - // trial vector y is refined with Rosetta stage s to y')
    - **Doubled Population** $\leftarrow$ **Add(y')**
  - **for each Individual** $\mathbf{z} \in$ **Doubled Population**
    do $\mathbf{f_{shared}(z)} \leftarrow$ **Shared Fitness**$(\mathbf{f_{original}(z)})$
    // f_{shared} is calculated for each z of the doubled population
  - **Sorted Doubled Population** $\leftarrow$ **Sort(Doubled Population)**
  - // The doubled population is sorted according to shared fitness
  - **Population** $\leftarrow$ **Select the best half part(Sorted Doubled Population)**
  - // The worst half part of the enlarged population is removed
  - // to define the new population in next generation.
  - // Elitism preserves the overall best found solution if that best
  - // individual is removed during this selection process.

poorer individuals, which can make the evolutionary progression of the population more difficult.

Therefore, the individuals of the population are classified into different groups (species) according to their similarity. As explained by Li et al. [64], *SDE* complexity is $O(N)$ in the best case and $O(N^2)$ in the worst case. The main advantage of speciation is its ability to maintain high diversity and stable niches over generations, while the main disadvantage is the selection of the radius parameter r_s [45].

2.6. Structural diversity measure

The protein structural diversity measure defined by Garza-Fabre et al. [31] was used. This structural measure defined by those authors describes (coarsely) the relative position of each pair of Secondary Structure Elements (SSEs) with respect to each other. Distances are computed between the C_α atoms of the amino acid residues at the center of the SSEs, as Fig. 3 illustrates. For a given protein conformation with E SSEs, the set of interdistances between each pair of SSEs is calculated (the number of interdistances is $\binom{E}{2}$ for a protein with E SSEs). Each interdistance is normalized considering half of its maximum distance when the protein conformation is fully extended (as indicated by Garza-Fabre et al. [31], practically all the explored conformations in their study corresponded to values within 50% of that upper bound). Such SSEs are determined, for a given protein sequence, with a predictor (PSIPRED [67]), a process that is prior to the evolutionary search.

Finally, to measure the structural difference between two protein folded conformations, the Root Mean Square Error (RMSE) between the sets of interdistances of the two proteins is calculated (Garza-Fabre et al.

Algorithm 2.3: Algorithm for determining species seeds in ***SDE***

// Input: L_{sorted} - a list of all the individuals sorted in decreasing fitness values
// Output: S - a list of all dominating individuals identified as species seeds
$S = \emptyset;\ counter = 0;\ s = 1$
$Get\ best\ unprocessed\ p\ \epsilon\ L_{sorted}\ as\ the\ seed\ of\ the\ first\ species\ s1\ (S \leftarrow p)$
$i = 2$; // i refers to the index in L_{sorted}
for each $s \in$ Number of species - 1
do
- **while** $(distance(i,p) \leq r_s)$ & $(counter < Species\ Size)$
  - **do**
    - i=i+1
    - counter=counter+1
- $p \leftarrow i$
- $seed\ of\ specie\ s+1 \leftarrow p$
- $(S \leftarrow S \cup p)$
- $counter = 0;\ i = i+1;\ s = s+1$

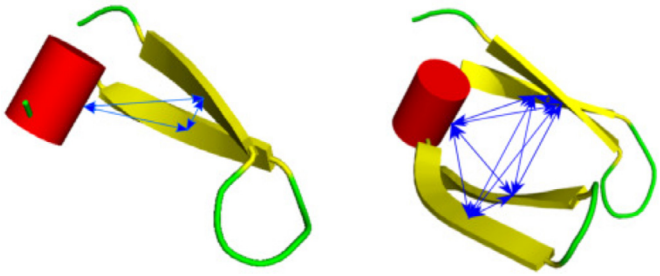

Fig. 3. Interdistances between pairs of secondary structure elements in hypothetical proteins. Left: protein with 2 strands and 1 helix - 3 interdistances between SSEs. Right: protein with 4 strands and 1 helix - 10 interdistances between SSEs. The interdistance set coarsely describes the protein conformational fold.

[31] considered, for this final measure, the interdistances between the start and final points of the SSEs). This simple procedure is appropriate for the purpose of the comparison of the folds of two proteins, and does not have the problems of calculating the best superposition or alignment between two proteins when the RMSD (Root Mean Square Deviation) between the atom positions of the two proteins is considered as measure of structural difference. This final RMSE is used for measuring the distance between encoded protein conformations in the niching methods.

2.7. Protein sequences

In the experiments, 30 different PDB proteins were used, the same employed in [31], in order to facilitate comparisons with previous studies that used algorithmic solutions to enforce diversity of folds in genetic populations. The main features of these proteins are set out in Table 2. The fragment libraries correspond to the first set of fragments used in [32].

3. Results

3.1. Results with CrowdingDE

3.1.1. Setup

The experiments with *CrowdingDE*, using the set of PDB proteins specified above, are designed to draw a comparison with the Rosetta ab initio protocol for obtaining a diversified set of protein folds. The Rosetta ab initio protocol is selected for comparison with all memetic approaches as it is a successful and widely used approach to address the PSP optimization problem when looking for protein structures with minimal energy. The experiments are also selected to illustrate the capabilities of the incorporation of niching for the hybrid evolutionary algorithm in the present problem.

In the experiments, regarding the DE algorithm, the DE strategy *DE/rand/1/bin* was used, which selects the base vector x'_1 (Algorithm 2.1) randomly and which implies the lowest selective pressure. The parameter values of DE genetic operators are: $CR = 0.99$ and $F = 0.025$. These values were experimentally tuned to provide the best results for most proteins, that is, for providing the best energy results in the given number of generations (and corresponding energy/fitness evaluations). The reason for the low value of F is that it generates small perturbations in the dihedral angles of the mutant vectors. In the same sense, the high CR value ensures few changes in the final trial vector with respect to the mutant vector, so the trial vectors are small variations of the base vector. On the contrary, with large variations, there would be many conflicts (atoms in the same position) in the resulting candidate vectors.

Moreover, the comparison of results must imply that both approaches (*CrowdingDE* and Rosetta ab initio) use the same number of fragment insertion attempts (which require an energy test per attempt). Consequently, both methods would use the same number of energy calculations. In the evolutionary algorithm there are no extra fitness calculations, since the fitness is given by the final energy calculation in the final fragment insertion attempt in the encoded protein of the genetic population or in the candidate vectors.

In the case of Rosetta, for each target protein, Rosetta was run to generate a set of 1000 candidate conformations (the same number used in [31]), given its stochastic nature in each run. This number of runs means that the obtained RMSD (from the native structure) distributions do not vary significantly between sets of runs [31]. That is, the Rosetta ab initio protocol was run 1000 times, where each run produced a single final conformation or decoy. Rosetta recommended parameter settings [25] were considered in all runs. Moreover, as in [31], the Rosetta parameter *increase_cycles* was set to 10; that is, the default values of cycles (fragment insertion attempts) in the different Rosetta stages (Table 1 in Section 2.1) are multiplied by that value.

In *CrowdingDE*, a population of 100 individuals was used (as in Garza-Fabre et al. [31] with their hybrid GA), and 10 independent runs

Table 2
PDB proteins used in the experiments. The columns correspond to the protein PDB Id, its amino acid number, and its native fold topology.

PDB	Size	Fold topol.	PDB id	Size	Fold topol.	PDB id	Size	Fold topol.
1acf	125	$\alpha-\beta$	1bgf	118	α	1bkrA	108	α
1c8cA	62	$\alpha-\beta$	1c9oA	66	β	1cg5B	141	α
1ctf	68	$\alpha-\beta$	1dhn	121	$\alpha-\beta$	1elwA	117	α
1eyvA	131	α	1fna	91	β	1gvp	87	β
1hz6A	61	$\alpha-\beta$	1iibA	103	$\alpha-\beta$	1kpeA	108	$\alpha-\beta$
1lis	125	α	1npsA	88	$\alpha-\beta$	1opd	85	$\alpha-\beta$
1rnbA	109	$\alpha-\beta$	1ten	89	β	1tig	88	$\alpha-\beta$
1tit	89	β	1tul	102	$\alpha-\beta$	1vcc	77	$\alpha-\beta$
1who	94	β	1wit	93	β	256bA	106	α
2chf	128	$\alpha-\beta$	2ci2I	62	$\alpha-\beta$	2vik	122	$\alpha-\beta$

of the hybrid evolutionary algorithm were used for generating the same 1000 final solutions[2] In *CrowdingDE*, the fragment insertion attempts are applied to the candidate individuals to refine them to better conformations and to the encoded conformations of the population in generation 1 of the three evolutionary stages (Algorithm 2.1), and the evolutionary algorithm is run over 100 generations in those three sequential stages. Consequently, the number of fragment insertion attempts for refining each candidate solution/encoded conformation must be limited. The total number of fragment insertion attempts is the same in both approaches, since Rosetta used *increase_cycles*=10, and the evolutionary algorithm used *increase_cycles*=0.1, but with 100 individuals and over 100 generations working with the same Rosetta stages, evolutionary process which was repeated 10 times. In other words, each of the 1000 solutions/DE candidates considered in the pool of solutions, uses only 1% of the Rosetta fragment insertion attempts, since the same process is repeated in 100 generations.

3.1.2. Comparison with Rosetta ab initio

Fig. 4 shows a comparison of the results of Rosetta against the results of *CrowdingDE*, using two values for the parameter CF, which determines the percentage of individuals of the population that are considered when DE trial individuals are compared with their similar ones in the population (Algorithm 2.1). Additionally, to test the capability of *CrowdingDE* to maintain diversity in the population without loss of genetic variability, the comparison includes the results of 10 hybrid DE runs without using crowding (the trial vector is compared with its corresponding target as in standard DE, Algorithm 2.1), again using the lowest selective pressure (scheme *DE/rand/1/bin*). Thus, the results with hybrid DE without crowding allow us to see the effect of the inclusion of the niching technique. Fig. 4 includes the results with half of the proteins, whereas Supplementary Material includes the figures with all the proteins.

Fig. 4 shows the normalized energy value (*score3*) on axis y, whereas on axis x the RMSD value (from the native structure) is shown. These standard graphs in PSP provide the information necessary to assess the distribution of distances (RMSD) of the optimized protein conformations, together with the optimization (in energy terms) obtained in those final solutions. The *score3* energy values were normalized between 0 and 1, taking into account the lowest and highest values obtained by the 4 considered approaches. Note that the PDB proteins act as benchmarks, since the native structure is known. Note also that the objective is to obtain folds close to the native structure (RMSD close to 0), and as diverse as possible, given the inaccuracies of the energy landscape, which will be described in what follows.

Clearly, *CrowdingDE* improves the energy values of the solutions obtained in all proteins with respect to Rosetta. At the same time, *CrowdingDE* maintains a diverse set of folds in the final populations, even with the energy improvement in most solutions compared to the Rosetta solutions (a high improvement in the average energy in proteins like *1hz6A, 1kpeA, 1lis* and *256bA*[3]). The lowest RMSD values are similar in both approaches (*CrowdingDE* and Rosetta) in most proteins, except for protein *1kpeA*, in which *CrowdingDE* clearly outperforms Rosetta. The main difference between the two approaches with *CrowdingDE* is that, with $CF = 10\%$, the energies obtained, in most proteins, are better (with respect to $CF = 100\%$), especially considering the average energy. This is logical since, with $CF = 10\%$, several solutions can fall in the same landscape area, which contributes to a better exploitation and to such better (or slightly better) energy results.

In hybrid DE without crowding, the final solutions (in each of the 10 independent runs) tend to be located in the same area, even with the low selective pressure applied. The increased exploitation (with respect to *CrowdingDE*) implies that the best energy values are similar (or even slightly better) with respect to the ones obtained with *CrowdingDE*. Regarding RMSD distributions, without crowding, hybrid DE clearly generates worse values in terms of RMSD dispersion with respect to the previous runs with *CrowdingDE*, and also with respect to Rosetta ab initio. The loss of genetic variability in the hybrid DE runs (without niching) implies that the population is easily concentrated in the different runs.

Tables S1/S2 (in Supplementary Material) show the best RMSD/energy values obtained with the different approaches. Nevertheless, even with the clear energy improvement in the conformations obtained with *CrowdingDE*, the RMSD results (from the native structure) are not necessarily better compared to the final Rosetta solutions with worse energies. However, *CrowdingDE* improves the energy without focusing on a particular area of the conformational space, as the RMSD dispersions in Fig. 4 show. Three patterns can be seen regarding the energy/RMSD distribution plots:

i The areas of better energy in proteins like *1fna, 1hz6A, 1kpeA, 1rnbA* and *1tig* tend to correspond to folds closer to the native structure.
ii Several proteins (such as *1c9oA, 1ctf, 1dhn, 1eyvA* and *1ten*) present areas with clearly different local minima and different RMSD distances to the native structure.
iii Contrary to the first case (*i*), proteins such as *1lis, 1opd, 1tul* and *1wit* are examples of the deceptive nature of the Rosetta energy landscape, where the area of low energy values does not correspond to the conformations closest to the native structure.

A comparison with the values reported in [31] is difficult, since the authors used different fragment libraries in their Rosetta ab initio runs,

[2] Typical computing times are 55 min for each of the parallelized 10 independent *CrowdingDE* runs (protein 1c9oA as target). The experiments were run in the Supercomputing Center of Galicia (www.cesga.es), with Intel Xeon E5-2680 v3 processors at 2.50GHz, each one with 12 cores (24 threads) and 1GB of RAM.

[3] The tables in Supplementary Material include information with the best values, average values, and standard deviation regarding energy values and RMSD (from the native structure) in the different experiments.

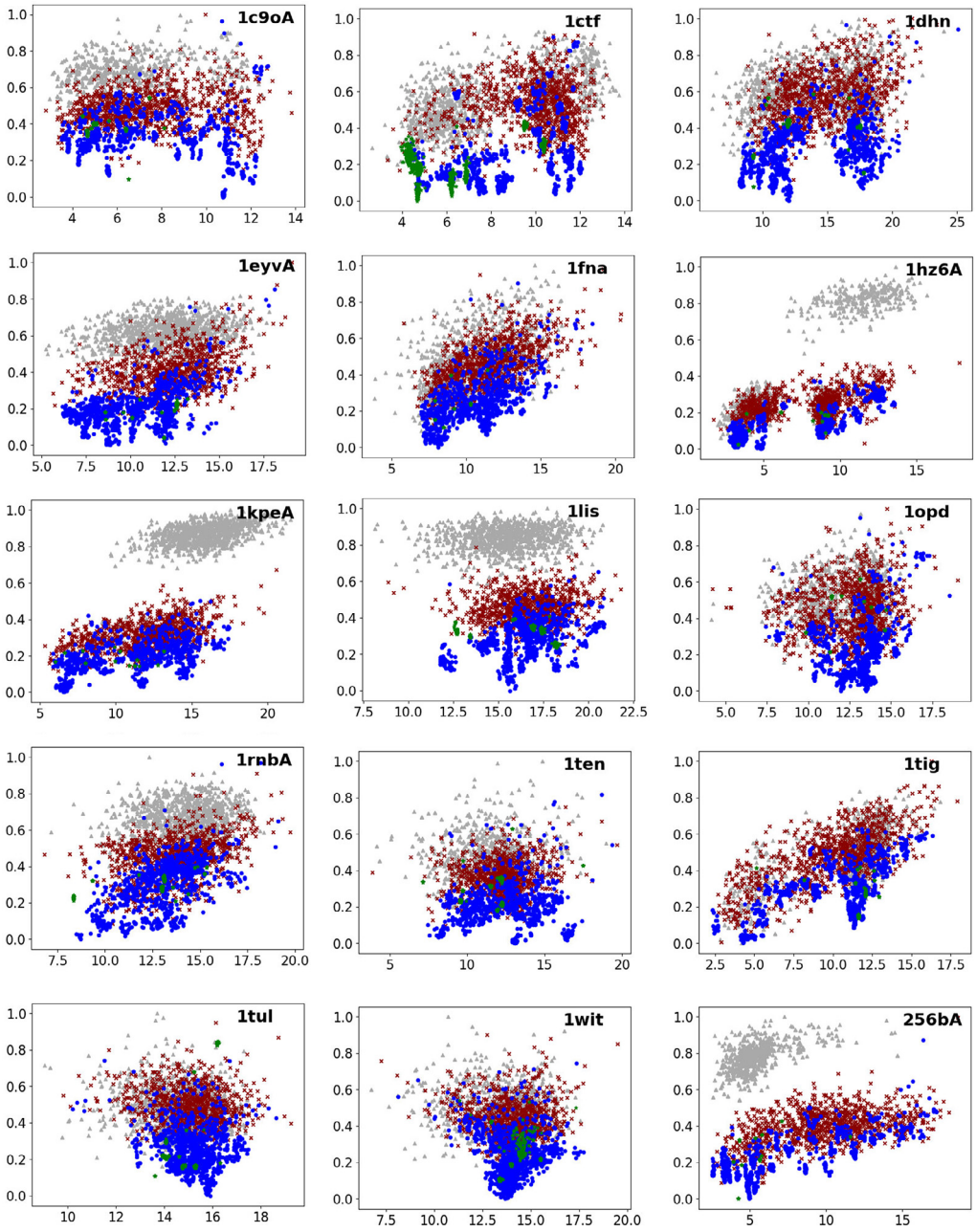

Fig. 4. Scaled energy vs. RMSD (from the native structure, in Å) with PDB proteins of Table 2. Gray: Rosetta results, Red: *CrowdingDE* (CF=100%), Blue: *CrowdingDE* (CF=10%), Green: Hybrid DE (without niching).

and the RMSD and energies that can be obtained depend to a large extent on that library (as remarked in [32]). Nevertheless, *CrowdingDE* obtains wide RMSD distributions independently of the protein. Note that this is the aim with niching, since the diversity of folds implies wide RMSD distributions. On the contrary, the RMSD distributions with the memetic algorithm of Garza-Fabre et al. [31] depend to a large extent on the GA variant used. For example, with protein *1hz6A*, the RMSD standard deviation values reported in [31], with 3 strategies/GA variants, are 0.34, 2.02 and 2.76, whereas the standard deviations of the final solutions with *CrowdingDE* are 2.69 ($CF = 100\%$) and 3.47 ($CF = 10\%$). With protein *1kpeA, CrowdingDE* RMSD standard deviations are 2.58 ($CF = 100\%$) and 3.02 ($CF = 10\%$), while the values in [31] are 1.03, 2.30 and 2.74, and with protein *1opd*, the values of the GA variants [31] are 0.38, 0.90 and 1.65, while with *CrowdingDE* are 2.03 ($CF = 100\%$) and 1.61 ($CF = 10\%$) (Table S1 in Supplementary Material).

3.1.3. Comparison with other approaches

The protocols defined by Kandathil et al. [32] were also used for comparison with *CrowdingDE* and Rosetta ab initio. These protocols, Iterated Local Search (ILS) and bilevel protocol [32], are a modification of Rosetta ab initio in stages 2 and 3. As noted in the Introduction, their bilevel protocol allows “Perturbation steps” in (predicted) loop regions, while “LocalSearch” steps are applied in the rest of the protein. The reason for performing more exploration in loop regions is that fragment libraries are less enriched for native-like structural features in those loop regions [32]. The Perturbation steps can perform large moves in conformational space, since the perturbed structure is accepted regardless of

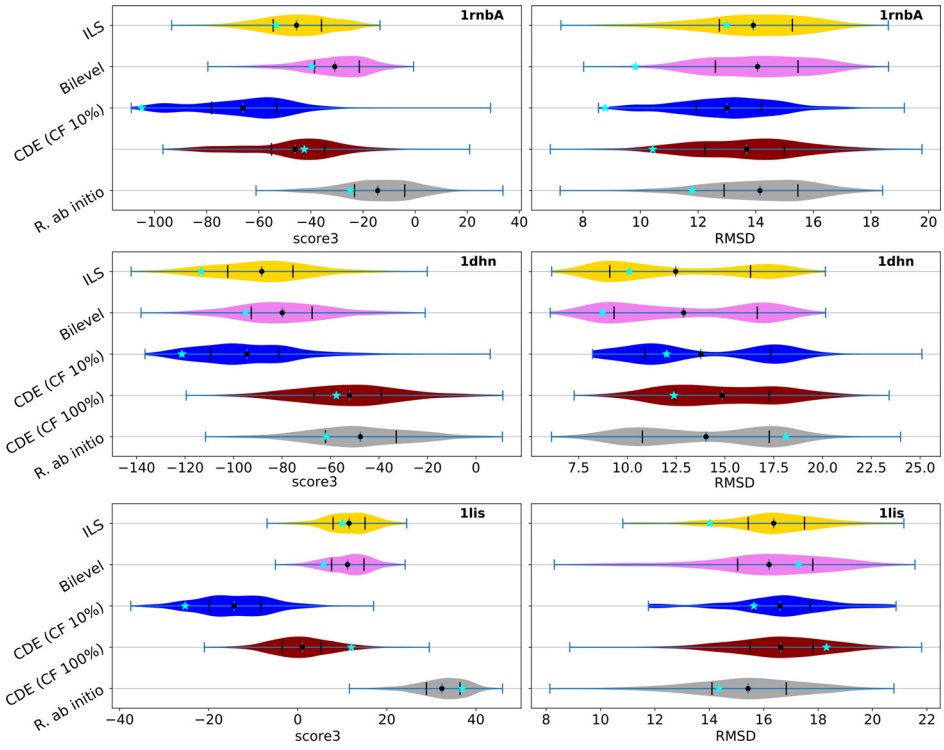

Fig. 5. Violin plots of energy (*score3*) distribution (left) and violin plots of RMSD (from the native structure, in Å) distribution (right) of the solutions with 5 different strategies. Upper figure: protein *1rnbA*, without a deceptive landscape; Figure in the middle: protein *1dhn*, with a multimodal landscape. Bottom figure: protein *1lis* with a deceptive landscape. Yellow: Iterated Local Search (ILS) [32], Pink: Bilevel protocol [32], Red: *CrowdingDE* (CF=100%), Blue: *CrowdingDE* (CF=10%), Gray: Rosetta ab initio. Light blue marks in violin plots of energy/RMSD distributions: solutions with the best RMSD/energy.

energy changes (contrary to the LocalSearch). In ILS both steps are iteratively applied in the whole protein: in one ILS iteration, one or more perturbations are applied, and subsequent fragment insertion attempts are performed using what is now greedy optimization. The code of both protocols was used to run these alternatives with the same fragment library employed in the other approaches and for obtaining once again 1000 final solutions using the same number of fragment insertion attempts.

Using the same setup specified by the authors [32], trajectories of the bilevel and ILS protocols were run with the parameter *increase_cycles* set to 100. Moreover, both protocols use an external archive to store 10 of the lowest-scoring structures in stages 2 and 3. In both protocols, the final stage (4) is applied to each archived structure and, consequently, the length of stage 4 is reduced by a factor of 10 [32]. 100 independent runs were applied with both protocols. Since the archiving strategy stores 10 of the lowest-energy solutions seen during each run, a set of 100 runs with each of the protocols returns 1000 decoy structures in total.

Therefore, the number of scoring function evaluations used per decoy is the same in these protocols (bilevel and ILS) and in Rosetta ab initio (with the Rosetta ab initio parameter setup explained previously). As we noted above, *CrowdingDE* runs also use the same number of scoring function evaluations for obtaining the evolved 1000 solutions.

Three proteins were selected to illustrate the results of the different approaches regarding their search behavior in different protein energy landscapes, although tables S1 and S2 in Supplementary Material include the results with the whole protein set (as well as figures with more proteins). Fig. 5 include the results with those proteins, showing the distributions (violin plots) of energy and RMSD of the final 1000 solutions with *CrowdingDE* and the Rosetta-based protocols. The (light blue) marks in the violin plots of the RMSD distribution show where the best solution (for each search alternative) is in energy terms and, likewise, the same marks in the violin plots of energy distribution show where the best solution is with respect to RMSD. The different behaviors of the distributions correspond to the same categorization used previously:

i Fig. 5 includes an example of a protein (*1rnbA*) with an energy landscape which is not deceptive, in the sense that obtaining better energies implies that better RMSDs are also obtained. The (light blue) marks in the violin plots show the tendency that the best solution in energy terms is close to the best solution in RMSD terms and vice versa.
Note again that the best approach (in energy terms) is *CrowdingDE* with CF=10%, since it obtains the best energy results compared with the other approaches considered (as can clearly be seen with the average energy of the final solutions). However, even with the tendency of the correlation between better energies and better RMSDs, this does not guarantee that *CrowdingDE* (CF=10%) obtains the best results in RMSD terms.

ii The final solutions in protein *1dhn* tend to be located in different areas that correspond to different local minima (Fig. 5).
In the violin graphs, with protein *1dhn*, it can be seen how the solutions are concentrated mainly in two local minima and with all the approaches (as shown in the RMSD distributions). Moreover, in the violin plots of RMSD distribution, it can be seen how the best solutions in energy terms (light blue marks) can be located in different areas of these local minima.

iii The Rosetta energy landscape in proteins like *1lis* is clearly deceptive (Fig. 5). *CrowdingDE* approaches present a clear energy improvement with respect to the Rosetta-based protocols and Rosetta ab initio, and at the same time their optimized solutions tend to change to distant folds of the native state, as the RMSD distributions show.
It should be noted that Rosetta presents the best results in RMSD terms (considering the average RMSD of the solutions), while the best approaches in energy terms (*CrowdingDE*) correspond to the worst approaches considering the average RMSD. This shows the clear deceptiveness of the energy landscape, where an improvement in energy tends to move the solutions to areas far from the native structure. The light blue marks in the violin plots also show, for most approaches, the lack of correlation between better energies and better RMSDs.

Figures S2a, S2b and S2c in Supplementary Material (Additional Figures) contain more examples of these three categories regarding the nature of the protein energy landscape.

Regarding the energies obtained, *CrowdingDE* is clearly the best approach, in most proteins, to sample the energy landscape in order to obtain the best energy optimized solutions that correspond with areas of different minima. The violin distributions in Fig. 5, as well as the results shown in Table S2 in Supplementary Material with all proteins, show how the *CrowdingDE* alternatives clearly outperform the others in terms of energy. In particular, *CrowdingDE* (CF=10%) generally obtains the best results in energy terms, in most cases considering the average energy and also the best (minimized) energy. The Rosetta-based modified protocols of Kandathil et al. [32] also provide better energies in their final solutions with respect to Rosetta ab initio in practically all proteins.

With respect to the RMSD distributions, there is no one approach that outperforms the others in all proteins. The comparison between the Rosetta-based protocols of Kandathil et al. [32] shows that, in most proteins, ILS provides better performance than the bilevel protocol, in terms of energy and average RMSD, which agrees with the results specified in [32] with two different fragment library sets. The authors explain this by the fact that the bilevel protocol may be misguided by inaccurate secondary structure predictions (of the loop regions). Finally, *CrowdingDE* solutions tend to present worse RMSD distributions only in proteins with a clear deceptive fitness landscape (Fig. 5). In this case where the area of best energies tends to move away from the native structure, the niching effect is not enough to retain the folds close to the native structure that could be present in the genetic population during the evolutionary process.

3.2. Results with SharingDE

The DE version with fitness sharing (*SharingDE*, Algorithm 2.2 in Section 2.4) is here tested with the same setup, regarding the DE parameters ($CR = 0.99$, $F = 0.025$), as in the previous experiments. The DE strategy $DE/rand/1/bin$ was again used to define the DE trial vectors. Regarding the FS parameters, the value of the sharing level (α) in Eq. 2 was set to 1 (as in most applications with FS), while the sharing radius (σ_{share}) was set to different values.

Three proteins were selected to test the results with *SharingDE*. The same methodology for comparison with Rosetta ab initio from previous experiments was followed, with 10 independent runs of the *SharingDE* algorithm (population of 100 individuals), comparing the final populations to 1000 independent Rosetta ab initio runs.

Fig. 6 shows the results when *SharingDE* used five different values for the parameter sharing radius (σ_{share}, Section 2.4), in order to define the vicinity of each encoded conformation. The values of σ_{share} correspond to different orders of magnitude, from the highest value $\sigma_{share} = 0.001$ to the lowest value $\sigma_{share} = 0.00001$ (the measure of fold difference, defined in Section 2.6, is normalized and varies within a small range). Figures S3a and S3b in Supplementary Material (Additional Figures) include the results with more proteins.

The results with the proteins employed in Fig. 6 shows different aspects:

i With lower values for parameter σ_{share}, the energies obtained tend to be lower. This is logical since it is more difficult to find a conformation in the vicinity when the distance threshold defined by σ_{share} is decreased. Therefore, the evolutionary process tends to be similar to a standard DE evolution without niching. In fact, with the lowest value ($\sigma_{share} = 0.00001$), the best energies are not necessarily the best ones, which means that there is a premature convergence that the niching method does not avoid.
On the contrary, with the highest value in σ_{share}, the wide neighborhood for each conformation means that most conformations have a large number of neighbors during the evolutionary process, which makes the progression towards the local minima more difficult, presenting average energy values even worse with respect to Rosetta in most proteins (detailed values in Table S4 in Supplementary Material).
Fig. 7 illustrates this last effect, showing the fitness evolution in particular runs of *SharingDE* with protein *1c9oA* and 3 different σ_{share} values. With the largest value (left part of Fig. 7), there are persistent fluctuations in the average fitness, since several individuals can worsen their (effective) fitness if these fall in the same close area. In contrast, with the lowest value of σ_{share}, there is more exploitation, as can be seen by the fact that the average fitness is closer to the best fitness. Note that the fitness of the best individual does not present fluctuations, since elitism of the best individual was used (Algorithm 2.2, Section 2.4).

ii The concentration of the population in fewer local minima, when σ_{share} decreases, tends to decrease the RMSD dispersion of the final solutions.
However, even with very low values of the defining parameter, the incorporation of niching allows the population to cover different local minima, as shown in the selected proteins.

iii Therefore, the most important decision with *SharingDE* is the value for σ_{share}, since it requires prior knowledge of the fitness landscape. With that knowledge, an appropriate value would be the one that separates adjacent local minima. Since this knowledge is not known in advance, it is also the main drawback of this niching method (in addition to its computational complexity).

The comparison with the results of *CrowdingDE* in Fig. 4 with the selected proteins shows that *CrowdingDE* presents a more continuous distribution (in RMSD terms) than the solutions optimized with *SharingDE*. The continuous distributions are present in the solutions with *SharingDE*, but with the highest value of σ_{share} and, consequently, without being optimized in energy terms. The same patterns regarding the energy/RMSD distribution in the solutions (discussed in Section 3.1) are present; for example, the different behavior with the deceptiveness in the landscape with protein *1wit* and the opposite behavior with protein *256bA* since, in the latter case, the minimization of energy moves the solutions towards the native structure. Protein *1c9oA* is an example of energy landscape with different local minima, as shown both in the distribution of solutions in Fig. 6 (with low values in σ_{share}) and Fig. 4 for the same proteins.

3.3. Results with species-based DE (SDE)

Finally, the species-based DE niching alternative (*SDE*), described in Section 2.5, was tested with a subset of 6 proteins.

Fig. 8 shows results with *SDE* using three different r_s values for defining the distance between the seeds of species (and the individuals associated with the species), as explained in Algorithm 2.3 in Section 2.5. The distance between folds is again calculated with the measure defined in Section 2.6. In the *SDE* runs, 8 species were used, all with the same number of associated individuals. In these runs, the population was 96 individuals[4], so every species has 12 individuals. Once the species are defined, each of the species is evolved with an independent (hybrid) DE run (Section 2.5). In these runs, 5 DE generations are used (*NUMBER_GEN* in Section 2.5). The setup of DE parameters is the same as in the previous case with *SharingDE*. Fig. 8 corresponds to the final populations (in generation 300) in 10 independent runs.

The energy threshold to decide whether a species individual is randomized was set to a small value (5). That is, if the difference of energy between a species individual and the species seed is lower than that

[4] The number of individuals (96), instead of 100 as in previous cases, is because it is a multiple of 8 species, with the same number of individuals (12) in every species.

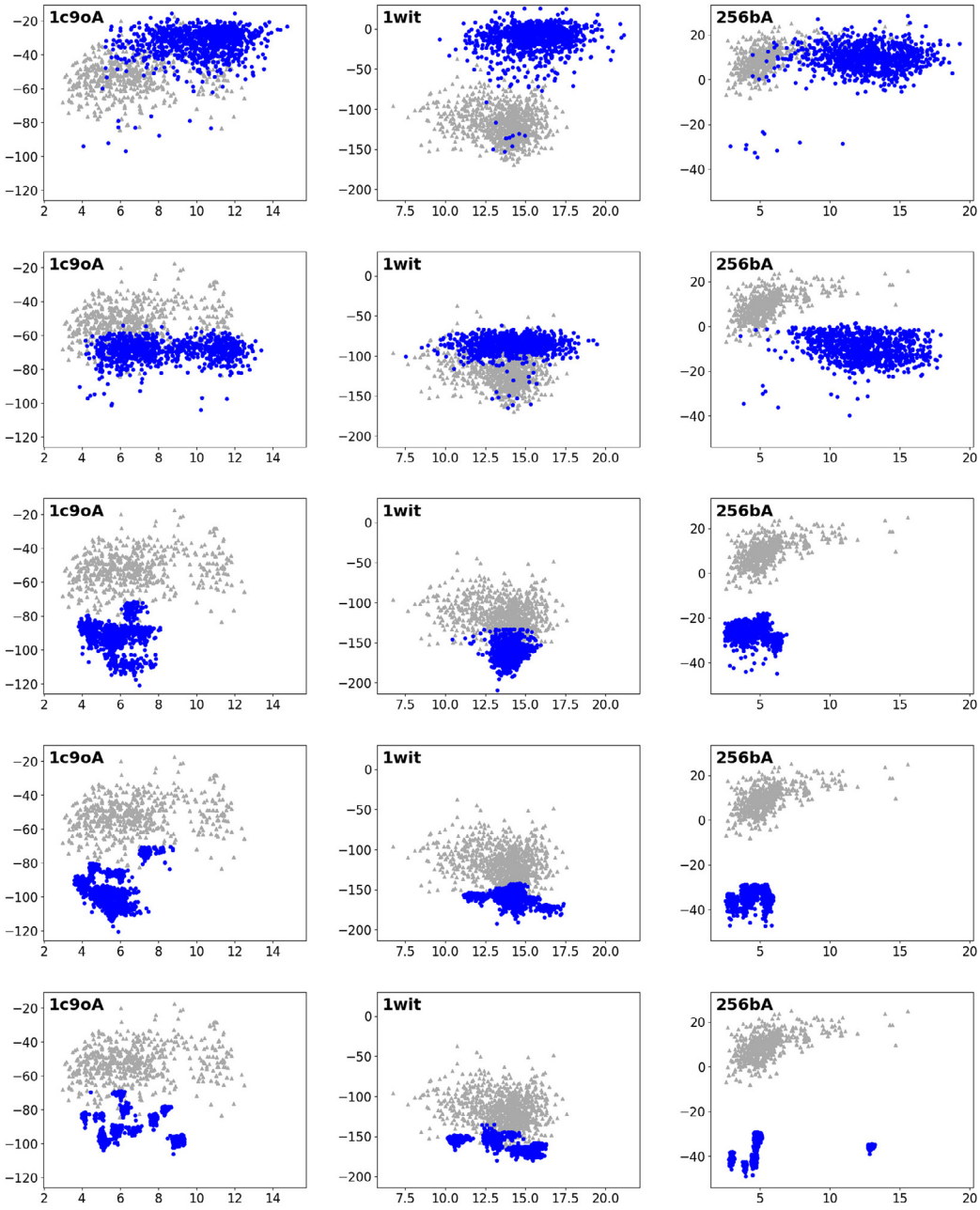

Fig. 6. Energy (*score3*, axis *y*) vs. RMSD (from the native structure, axis *x*, in À) for 3 proteins when *SharingDE* is used with 5 different values of the sharing radius (σ_{share}). From upper to bottom rows: $\sigma_{share} = 0.001$, $\sigma_{share} = 0.0005$, $\sigma_{share} = 0.0001$, $\sigma_{share} = 0.00005$ and $\sigma_{share} = 0.00001$. The Rosetta results of 1000 runs are shown in gray for comparison.

value, the individual is randomized (with the same general procedure to initialize the individuals of the initial population).

Each of the three evolutionary stages (with the corresponding Rosetta score) implies 100 generations. Therefore, there are 20 recalculations of the species seeds (and associated individuals) in those 100 generations (once the DE runs of each of the species is finished after 5 generations), and a total number of 60 recalculations in the whole evolutionary process.

Wide RMSD distributions are once again obtained in the final populations. With the lowest r_s value used (right subfigures in Fig. 8) there are some individuals with higher energy with respect to the larger r_s values. With a small r_s value the seeds can correspond to very close values (Algorithm 2.3 in Section 2.5). Consequently, with that lowest value in r_s and even with the low energy threshold (5) to decide if an individual in a species is randomized, many individuals in final generations tend to be randomized by the functioning of *SDE* given the high exploitation around the seeds, which implies the higher energies in part of the population.

The distributions of the solutions present the same behavior as in the previous approaches (Sections 3.1 and 3.2) for the different proteins. The protein landscape of protein *1wit* is, once again, the one that presents a clear deceptiveness, since *SDE* concentrates the search in the best area in terms of energy minimization, area that tends to move the population away from the native structure. The opposite behavior is present, once more, with the energy landscapes of proteins *1kpeA* and *256bA* that, even with their multimodal nature, the best energy area also tends to correspond to the area closest to the native structure. The other three proteins, *1c9oA, 1elwA* and *1hz6A*, present energy landscapes with

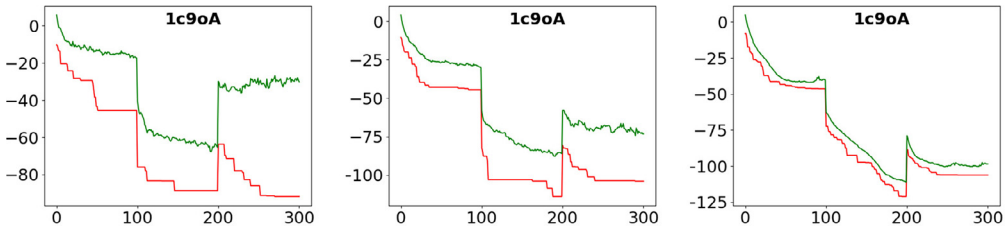

Fig. 7. *SharingDE* fitness evolution through generations and the 3 stages of the evolutionary algorithm (Section 2.2), using protein *1c9oA* as target and 3 different radii. Left - $\sigma_{share} = 0.001$, Center - $\sigma_{share} = 0.0005$, Right - $\sigma_{share} = 0.00001$. Green line - average fitness, Red line - best fitness.

different local minima with different depths, without a clear area in energy terms in which to concentrate most of the solutions.

Finally, Figure S4 in Supplementary Material illustrates the same effect of r_s values with two target proteins, but when no randomized individuals are incorporated into the subpopulations of species. In addition, Figure S5 in Supplementary Material shows the effect on the results of the number of species using the previous 6 proteins.

4. Discussion and conclusions

Anfinsen's dogma states that the amino acid sequence acts as the blueprint for determining protein folding to its native structure, and that the native conformation corresponds to the minimum of the Gibbs free energy [2]. Therefore, the protein structure prediction problem can be treated as an optimization problem, since the energy minimum of a defined energy model must be found to discover the conformation of the native structure. However, in atomic models of protein representation, a problem arises when that energy minimum does not necessarily correspond to the real native structure, as in the case of the Rosetta energy model, especially given its knowledge-based nature.

Given the ruggedness and deceptiveness of the Rosetta energy landscape, as well as the known inaccuracies in its energy model which make it difficult to correctly distinguish between the quality of folds in different local optima [31], one possibility to address the problem is to obtain a diversified set of decoys or conformations with different folds. As stated by Akhter et al., "It is highly desirable for the decoy generation stage to obtain an unbiased and uniformly-dense view of the landscape, so that obtained decoys cover the multitude of basins possibly present in a protein energy landscape and not miss basins containing native conformations" [68]. And also, as stated by Garza-Fabre et al., [31] "it seems that a successful search technique for fragment assembly will have to incorporate improved mechanisms to generate and retain low-energy structures that correspond to distinctly different folds".

With that goal in mind, we integrated niching methods (crowding, fitness sharing and speciation) into a hybrid combination between an evolutionary algorithm (DE) and a local search based on the fragment insertion technique. Unlike previous studies with the same aim, which analyzed the effect of different genetic operators for obtaining diverse folds, the integration of niching allows us to straightforwardly obtain such a desired and diverse set of folds in the optimized solutions of the genetic populations. The first benefit of using niching is the most difficult premature convergence of the evolutionary algorithm. As noted by Li et al., [44], "Seeking multiple good solutions in different regions of the search space may help with keeping a diverse population, counteracting the effect of genetic drift". Nevertheless, this is not the main benefit in the current problem, since the most interesting aspect is to simultaneously obtain a population of optimized solutions corresponding to different folds.

In this integration of niching, the structure of the evolutionary process followed the previous study by Garza-Fabre et al. [31], in which the individuals of the population were optimized through different generations that correspond to the different Rosetta ab initio phases. The evolutionary process presents three different phases, and in each of these evolutionary phases we used the Rosetta fitness score, cycles of fragment insertion attempts, and fragment lengths of the corresponding Rosetta phase. This means that the solutions of the population can be progressively refined with the same methodology of Rosetta ab initio, and also allows for a direct comparison with Rosetta ab initio results under the same number of fragment insertion attempts. Moreover, since niching methods rely on a distance measure between individuals, protein conformations in the current problem, a measure for calculating the distance (structural difference) between two folds has to be considered. The measure defined by Garza-Fabre et al. [31] was employed in the present study, which takes into account the interdistances of the secondary structure elements in order to describe the conformational fold.

From the experiments performed with the niching methods considered, several conclusions can be drawn:

i The hybrid evolutionary algorithm (with and without the integration of niching methods) performs better for sampling the energy landscape to obtain optimized solutions in energy terms, compared to Rosetta ab initio and to two Rosetta-based protocols (ILS and bilevel protocols [32]), as shown with the energy distributions of final solutions in most proteins. This enhanced search capability of a memetic algorithm is not novel in evolutionary computation, but it demonstrates that, also in PSP approaches based on energy minimization, the combination of the global DE search and the local refinement capability of the fragment insertion technique is useful.
From this initial memetic combination between the fragment replacement technique and standard DE, the use of self-adaptive versions of DE [69] should be checked, with a strict comparison between DE variants [70,71]. In particular, versions such as L-SHADE [72] to overcome the parameter tuning, also integrating a neighborhood mutation [45] to further strengthen the niching effect.

ii With the incorporation of the niching methods, the solutions of the final generations of the evolutionary process present a diverse set of folds with different distances (RMSD) from the real native conformation. The solutions present wider RMSD distributions with respect to the use of the evolutionary algorithm without niching, obtaining conformations closer to the native structure (in RMSD values) in some proteins with respect to Rosetta ab initio.

iii Regarding the different niching methods employed, in the case of *CrowdingDE*, its main advantage is that it only needs a parameter in its implementation (crowding factor CF), which can control the selective pressure, as shown in the runs with different CF values.
The main drawback with fitness sharing in *SharingDE* was to set the parameter σ_{share}, which defines the vicinity of a conformation in order to recalculate its effective fitness. The ideal parameter value could be the distance that separates two adjacent local minima in the fitness landscape, which obviously requires prior knowledge of the landscape. The experimentation with different values of σ_{share} showed the high dependence of the results (final conformations) on the value of that parameter. Something similar can be said about the speciation niching method (*SDE*), which again requires the appropriate value for defining the vicinity of species around their seeds, as well as the tuning of the number of species. Moreover, premature convergence must be addressed in the small subpopulations of each species, as is done with the incorporation of randomized individuals. Since crowding does not require a parameter to decide a distance or threshold that separates local minima, *CrowdingDE* was found to be the most useful niching technique in the current problem, espe-

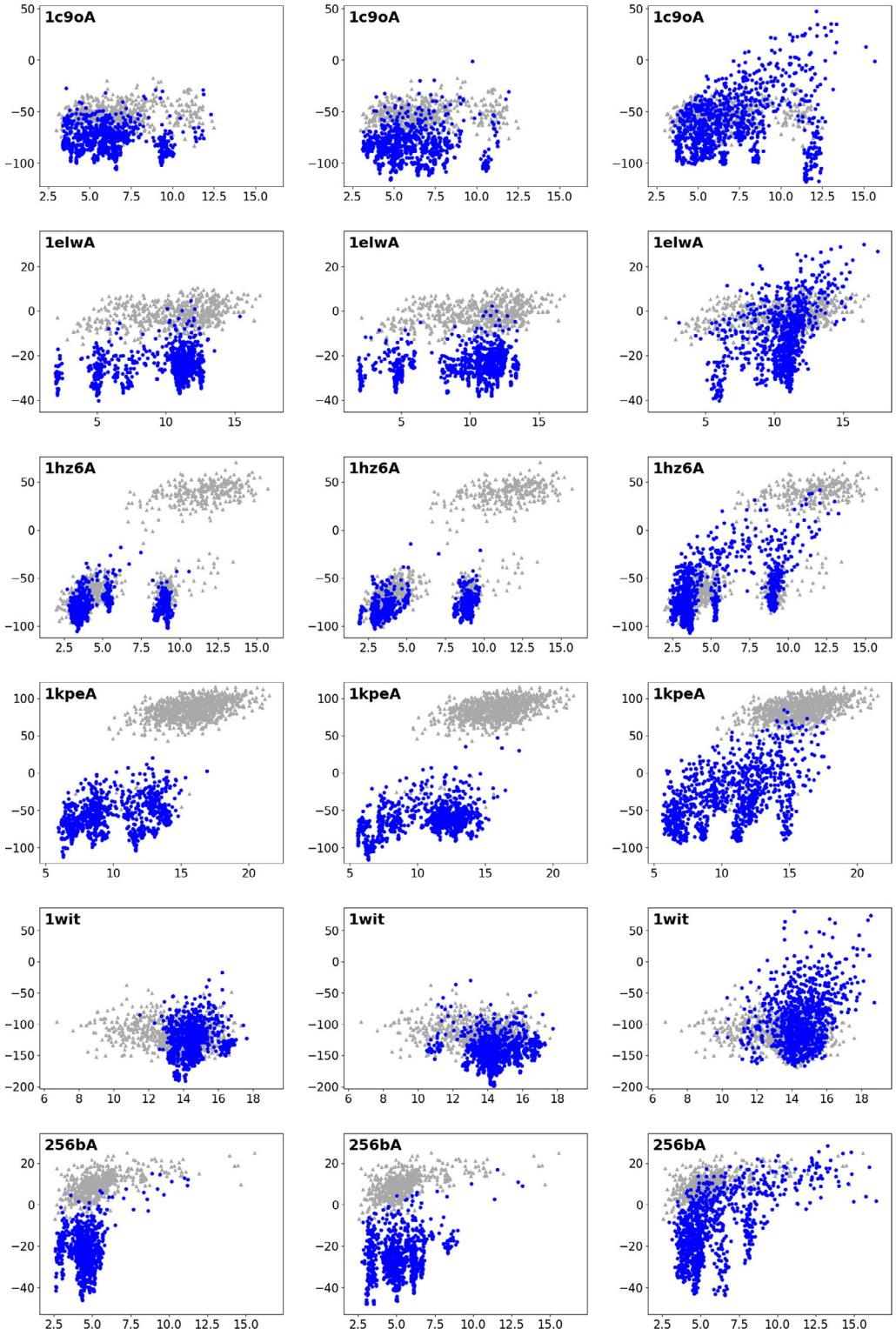

Fig. 8. Energy (*score3*, axis *y*) vs. RMSD (from the native structure, axis *x*, in Å) for 6 proteins and with 3 thresholds (r_s) for defining the species seeds in *SDE*. Left: $r_s = 0.05$, Center: $r_s = 0.01$, Right: $r_s = 0.001$. The Rosetta results of 1000 runs are shown in gray for comparison.

cially given the parameter decision process in the other two niching methods.

Other niching strategies should be explored as future work. For example, recent DE niching strategies based on niching competition [73]. Also, as noted above, an interesting approach is the integration of a neighborhood mutation into the DE process, as done by Qu et al. [45], to reinforce the niching effect. In neighborhood mutation, the generation of difference vectors is limited to nearby individuals. With its integration into standard niching methods, neighborhood mutation was able to induce stable niching behavior in DE, providing better and more consistent performance than other multimodal optimization algorithms on different benchmark functions [45]. There-

fore, the neighborhood mutation should also be explored in the PSP problem.

iv In the comparison of results of all the alternatives there is no one approach that outperforms the others in all proteins regarding the lowest RMSD. However, the Rosetta-based ILS and bilevel protocols [32] present the lowest RMSD or lowest average RMSD in more proteins than the other approaches, as the results show in the Supplementary Material tables. Consequently, an alternative worth exploring is the integration of crowding (and the other niching methods) with the greater explorative capability provided by the perturbation steps of the ILS/bilevel protocols.

v The results illustrate the large degree of deceptiveness in the Rosetta energy landscape for many proteins. The hybrid evolutionary versions defined here clearly present a better ability to sample the energy landscape with respect to the Rosetta-based protocols in order to find optimized solutions with minimized energy, but this does not guarantee the best results in terms of solutions closer to the native structure. This is explained by the deceptiveness of the landscape, in which the area of best energies moves away from the native structure.

In conclusion, then, the incorporation of niching represents a straightforward alternative to address the problem of deceptiveness in such protein energy landscapes, although each niching method has its own problems regarding computational complexity and parameter setup. The solutions obtained present wide RMSD distributions, although this does not resolve the problem with proteins with a clear deceptive energy landscape, such as the specific examples discussed in the experiments described here.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Daniel Varela: Methodology, Software, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing. **José Santos:** Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing, Project administration, Funding acquisition.

Acknowledgments

This study was funded by the Xunta de Galicia and the European Union (European Regional Development Fund - Galicia 2014–2020 Program), with grants CITIC (ED431G 2019/01), GPC ED431B 2019/03 and IN845D-02 (funded by the “Agencia Gallega de Innovación”, co-financed by Feder funds, supported by the “Consellería de Economía, Empleo e Industria” of Xunta de Galicia), and by the Spanish Ministry of Science and Innovation (project PID2020-116201GB-I00).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.swevo.2022.101062.

References

[1] A. Tramontano, Protein Structure Prediction. Concepts and Applications, Wiley-VCH, 2006.
[2] C. Anfinsen, Principles that govern the folding of proteins, Science 181 (96) (1973) 223–230.
[3] A. Senior, R. Evans, J. Jumper, et al., Improved protein structure prediction using potentials from deep-learning, Nature 577 (2020) 706–710, doi:10.1038/s41586-019-1923-7.
[4] A. Márquez-Chamorro, G. Asencio-Cortés, C. Santiesteban-Toca, J. Aguilar-Ruiz, Soft computing methods for the prediction of protein tertiary structures: a survey, Appl. Soft. Comput. 35 (2015) 398–410.
[5] N. Krasnogor, W. Hart, J. Smith, D. Pelta, Protein structure prediction with evolutionary algorithms, in: Proceedings GECCO’99 - Conference on Genetic and Evolutionary Computation, 1999, pp. 1596–1601.
[6] X. Zhao, Advances on protein folding simulations based on the lattice HP models with natural computing, Appl. Soft Comput. 8 (2008) 1029–1040.
[7] R. Unger, The genetic algorithm approach to protein structure prediction, Struct. Bond. 110 (2004) 153–175.
[8] G. Zhang, X. Zhou, X. Yu, X. Hao, L. Yu, Enhancing protein conformational space sampling using distance profile-guided differential evolution, IEEE/ACM Trans. Comput. Biol. Bioinf. 14 (6) (2017) 1288–1301.
[9] V. Cutello, G. Nicosia, M. Pavone, J. Timmis, Immune algorithm for protein structure prediction on lattice models, IEEE Trans. Evol. Comput. 11 (1) (2007) 101–117.
[10] S. Fidanova, 3D HP protein folding problem using ant algorithm, in: Proceedings of BioPS’06 International Conference, 2006, pp. 19–26.
[11] M. Garza-Fabre, E. Rodriguez-Tello, G. Toscano-Pulido, Comparative analysis of different evaluation functions for protein structure prediction under the HP model, J. Comput. Sci. Technol. 28 (5) (2013) 868–889.
[12] H. Lopes, M. Scapin, An enhanced genetic algorithm for protein structure prediction using the 2D hydrophobic-polar model, Lect. Notes Comput. Sci. 3871 (2006) 238–246.
[13] W. Patton, W. Punch, E. Goldman, A standard genetic algorithm approach to native protein conformation prediction, in: Proceedings of 6th International Conference on Genetic Algorithms, 1995, pp. 574–581.
[14] S. Shatabda, M. Newton, M. Rashid, A. Sattar, An efficient encoding for simplified protein structure prediction using genetic algorithms, in: Proceedings IEEE Congress on Evolutionary Computation - IEEE-CEC 2013, 2013, pp. 1217–1224.
[15] A. Shmygelska, H. Hoos, An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem, Bioinformatics 6 (2005) 30.
[16] F. Neri, C. Cotta, Memetic algorithms and memetic computing optimization: a literature review, Swarm Evol. Comput. 2 (2012) 1–14.
[17] C. Cotta, Protein structure prediction using evolutionary algorithms hybridized with backtracking, Lect. Notes Comput. Sci. 2687 (2003) 321–328.
[18] N. Krasnogor, B. Blackburne, E. Burke, J. Hirst, Multimeme algorithms for protein structure prediction, Lect. Notes Comput. Sci. 2439 (2002) 769–778.
[19] J. Santos, M. Diéguez, Differential evolution for protein structure prediction using the HP model, Lect. Notes Comput. Sci. 6686 (2011). 323–323
[20] M. Rashid, F. Khatib, M. Hoque, A. Sattar, An enhanced genetic algorithm for ab initio protein structure prediction, IEEE Trans. Evol. Comput. 20 (4) (2018) 627–644.
[21] N. Boumedine, S. Bouroubi, A new hybrid genetic algorithm for protein structure prediction on the 2D triangular lattice, Comput. Sci. Math. (2019) 1907.04190.
[22] B. Olson, K. De-Jong, A. Shehu, Off-lattice protein structure prediction with homologous crossover, in: Proceedings GECCO 2013 - Conference on Genetic and Evolutionary Computation, 2013, pp. 287–294.
[23] L. Corrêa, B. Borguesan, C. Farfán, M. Inostroza-Ponta, M. Dorn, A memetic algorithm for 3D protein structure prediction problem, IEEE/ACM Trans. Comput. Biol. Bioinf. 15 (3) (2018) 690–704.
[24] L. Corrêa, M. Dorn, A multi-population memetic algorithm for the 3-D protein structure prediction problem, Swarm Evol. Comput. 55 (2020), doi:10.1016/j.swevo.2020.100677.
[25] Rosetta, Rosetta system, (http://www.rosettacommons.org).
[26] C. Rohl, C. Strauss, K. Misura, D. Baker, Protein structure prediction using rosetta, Meth. Enzymol. 383 (2004) 66–93.
[27] CASP, Protein structure prediction center, (http://predictioncenter.org/).
[28] K. Kaufmann, G. Lemmon, S. DeLuca, J. Sheehan, J. Meiler, Practically useful: what the rosetta protein modeling suite can do for you, Biochemistry 49 (2010) 2987–2998.
[29] A. Shmygelska, M. Levitt, Generalized ensemble methods for de novo structure prediction, PNAS 106 (5) (2009) 1415–1420.
[30] S. Saleh, B. Olson, A. Shehu, A population-based evolutionary search approach to the multiple minima problem in de novo protein structure prediction, BMC Struct. Biol. 13 (1) (2013) S4.
[31] M. Garza-Fabre, S. Kandathil, J. Handl, J. Knowles, S. Lovell, Generating, maintaining, and exploiting diversity in a memetic algorithm for protein structure prediction, Evol Comput 24 (4) (2016) 577–607.
[32] S. Kandathil, M. Garza-Fabre, J. Handl, S. Lovell, Improved fragment-based protein structure prediction by redesign of search heuristics, Sci Rep 8 (1) (2018) 13694.
[33] D. Simoncini, T. Schiex, K. Zhang, Balancing exploration and exploitation in population-based sampling improves fragment-based de novo protein structure prediction, Proteins Struct. Funct. Bioinf. 85 (2017) 852–858.
[34] L. Corrêa, B. Borguesan, M. Krause, M. Dorn, Three-dimensional protein structure prediction based on memetic algorithms, Comput. Oper. Res. 91 (2018) 160–177.
[35] F. Custódio, H. Barbosa, L. Dardenne, A multiple minima genetic algorithm for protein structure prediction, Appl. Soft Comput. 15 (2014) 88–99.
[36] X. Wei, X. Zheng, Q. Zhang, C. Zhou, Improved niche genetic algorithm for protein structure prediction, in: Bio-Inspired Computing - Theories and Applications. BIC-TA 2015. Communications in Computer and Information Science, 562, 2015, pp. 475–492.
[37] K. Price, R. Storn, J. Lampinen, Differential Evolution. A Practical Approach to Global Optimization, Springer - Natural Computing Series, 2005.
[38] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[39] S. Das, P. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.

[40] V. Feoktistov, Differential Evolution: In Search of Solutions, Springer, NY, 2006.
[41] H. Lopes, R. Bitello, Differential evolution approach for protein folding using a lattice model, J. Comput. Sci. Technol. 22 (6) (2007) 904–908.
[42] D. Varela, J. Santos, Combination of differential evolution and fragment-based replacements for protein structure prediction, in: GECCO 2015 Proceedings Companion, Workshop Evolutionary Computation in Computational Structural Biology, 2015, pp. 911–914.
[43] S. Das, S. Maity, B. Qu, P. Suganthan, Real-parameter evolutionary multimodal optimization - a survey of the state-of-the-art, Swarm Evol. Comput. 1 (2) (2011) 71–88.
[44] X. Li, M. Epitropakis, K. Deb, A. Engelbrecht, Seeking multiple solutions: an updated survey on niching methods and their applications, IEEE Trans. Evol. Comput. 21 (4) (2017) 518–538.
[45] B. Qu, P. Suganthan, J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, IEEE Trans. Evol. Comput. 16 (5) (2012) 601–614.
[46] M. Epitropakis, V. Plagianakos, M. Vrahatis, Finding multiple global optima exploiting differential evolution's niching capability, in: Proceedings IEEE Symposium on Differential Evolution (SDE), 2011, pp. 1–8.
[47] R. Mukherjee, G. Patra, R. Kundu, S. Das, Cluster-based differential evolution with crowding archive for niching in dynamic environments, Inf. Sci. 267 (2014) 58–82.
[48] K. De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems, Doctoral Dissertation, University of Michigan, Ann Arbor, MI, 1975.
[49] R. Thomsen, Multimodal optimization using crowding-based differential evolution, in: Proceedings IEEE Congress on Evolutionary Computation, 2004, pp. 1382–1389.
[50] D. Varela, J. Santos, Crowding differential evolution for protein structure prediction, in: Proceedings International Work-Conference on the Interplay between Natural and Artificial Computation - IWINAC 2019, Lecture Notes in Computer Science 11487, 2019, pp. 193–203.
[51] D. Varela, J. Santos, Protein structure prediction in an atomic model with differential evolution integrated with the crowding niching method, Nat. Comput. (2020), doi:10.1007/s11047-020-09801-7.
[52] J. Holland, Adaptation in Natural and Artificial Systems, An Arbor MI: University of Michigan Press, Cambridge, MA, USA, 1975.
[53] D. Goldberg, J. Richarson, Genetic algorithms with sharing for multimodal function optimization, in: Proceedings 2nd International Conference on Genetic Algorithms, 1987, pp. 41–49.
[54] G. Yang, Z. Dong, K. Wong, A modified differential evolution algorithm with fitness sharing for power system planning, IEEE Trans. Power Syst. 23 (2) (2008) 514–522.
[55] X. Li, Efficient differential evolution using speciation for multimodal function optimization, in: Proceedings GECCO 2005 - Conference on Genetic and Evolutionary Computation, 2005, pp. 873–880.
[56] S. Kmiecik, D. Gront, M. Kolinski, L. Wieteska, A. Dawid, A. Kolinski, Coarse-grained protein models and their applications, Chem. Rev. 116 (2016) 7898–7936.
[57] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (6) (1953) 1087–1092.
[58] J. Lee, S. Wu, Y. Zhang, Ab initio protein structure prediction, in: From Protein Structure to Function with Bioinformatics, Springer-London, 2009, pp. 3–25.
[59] M. Sippl, Knowledge-based potentials for proteins, Curr. Opin. Struct. Biol. 5 (2) (1995) 229–235.
[60] PDB, Protein Data Bank, (http://www.wwpdb.org).
[61] A. Hagler, S. Lifson, Energy functions for peptides and proteins, II: The amide hydrogen bond and calculation of amide crystal properties, J. Am. Chem. Soc. 96 (1974) 5319–5327.
[62] D. Whitley, V. Gordon, K. Mathias, Lamarckian evolution, the Baldwin Effect and function optimization, Lect. Notes Comput. Sci. 866 (1994) 6–15.
[63] B. Sareni, L. Krähenbühl, Fitness sharing and niching methods revisited, IEEE Trans. Evol. Comput. 2 (3) (1998) 97–106.
[64] J. Li, M. Balazs, G. Parks, P. Clarkson, A species conserving genetic algorithm for multimodal function optimization, Evol. Comput. 10 (3) (2002) 207–234.
[65] A. Pétrowski, A clearing procedure as a niching method for genetic algorithms, in: Proceedings 3rd IEEE International Conference on Evolutionary Computation, IEEE, 1996, pp. 798–803.
[66] J. Horn, The Nature of Niching: Genetic Algorithms and the Evolution of Optimal, Cooperative Populations, Dept. Comput. Sci., University of Illinois at Urbana-Champaign, Urbana, IL, USA, 1997. Technical Report UIUCDCS-R-97-2000
[67] D. Jones, Protein secondary structure prediction based on position-specific scoring matrices, J. Mol. Biol. 292(2) (1999) 195–202.
[68] N. Akhter, W. Qiao, A. Shehu, An energy landscape treatment of decoy selection in template-free protein structure prediction, Computation 6 (2) (2018) 39.
[69] S. Das, S. Mullick, P. Suganthan, Recent advances in differential evolution - an updated survey, Swarm Evol. Comput. 27 (2016) 1–30, doi:10.1016/j.swevo.2016.01.004.
[70] J. Del Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P.N. Suganthan, C.A. Coello Coello, F. Herrera, Bio-inspired computation: where we stand and what's next, Swarm Evol. Comput. 48 (2019) 220–250, doi:10.1016/j.swevo.2019.04.008.
[71] E. Osaba, E. Villar-Rodriguez, J. Del Ser, A.J. Nebro, D. Molina, A. LaTorre, P.N. Suganthan, C.A. Coello Coello, F. Herrera, A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems, Swarm Evol. Comput. 64 (2021) 100888, doi:10.1016/j.swevo.2021.100888.
[72] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6–11, 2014, IEEE, 2014, pp. 1658–1665, doi:10.1109/CEC.2014.6900380.
[73] W. Sheng, X. Wang, Z. Wang, Q. Li, Y. Chen, Adaptive memetic differential evolution with niching competition and supporting archive strategies for multimodal optimization, Inf. Sci. 573 (2021) 316–331, doi:10.1016/j.ins.2021.04.093.

Daniel Varela has a BSc (2012) and an MSc (2014) in Computer Science from the University of A Coruña (Spain). His PhD (2019) is from the Department of Computer Science and Information Technologies of the same institution. His research interests include evolutionary computation and computational biology. He is currently a postdoctoral fellow at the University of Lund (Sweden).

José Santos has an MSc in Physics (1989, specialization in Electronics) from the University of Santiago de Compostela (Spain), and a PhD from the same institution (1996, specialization in Artificial Intelligence). He is Full Professor in the Department of Computer Science and Information Technologies, University of A Coruña (Spain). His research interests include artificial life, neural computation, evolutionary computation, autonomous robotics, and computational biology.