Adaptive Stochastic Optimization to Improve Protein Conformation Sampling

Ahmed Bin Zaman, Toki Tahmid Inan, Kenneth De Jong, and Amarda Shehu.

Abstract—We have long known that characterizing protein structures structure is key to understanding protein function. Computational approaches have largely addressed a narrow formulation of the problem, seeking to compute one native structure from an amino-acid sequence. Now AlphaFold2 promises to reveal a high-quality native structure for possibly many proteins. However, researchers over the years have argued for broadening our view to account for the multiplicity of native structures. We now know that many protein molecules switch between different structures to regulate interactions with molecular partners in the cell. Elucidating such structures de novo is exceptionally difficult, as it requires exploration of possibly a very large structure space in search of competing, near-optimal structures. Here we report on a novel stochastic optimization method capable of revealing very different structures for a given protein from knowledge of its amino-acid sequence. The method leverages evolutionary search techniques and adapts its exploration of the search space to balance between exploration and exploitation in the presence of a computational budget. In addition to demonstrating the utility of this method for identifying multiple native structures, we additionally provide a benchmark dataset for researchers to continue work on this problem.

Index Terms—protein structure; conformation sampling, stochastic optimization.

1 Introduction

Understanding the three-dimensional/tertiary structure of proteins is central to the recognition of molecular partners in the cell [1] and so holds the key to obtaining a detailed understanding of protein function. A vast body of work in molecular biology has been devoted to protein structure determination. Early on, experimental techniques, such as X-ray crystallography, revealed static snapshots of protein molecules, capturing a protein in one tertiary structure. Motivated in part by the inability of X-ray crystallography to generalize over different protein molecules, computational approaches stepped in. They leveraged a narrow formulation of the protein structure determination problem, where the goal was the determination of a single structure, also referred to as the native structure, from a given protein amino-acid sequence [2]. Impressive computational advances instigated via the "Critical Assessment of protein Structure Prediction" (CASP) competition were made over the years [3]. In December 2020, they culminated in the AlphaFold2 method, which, contrary to what the name suggests, presented a major advance in protein structure determination (and not protein folding). Reports from CASP14 suggest AlphaFold2 can now obtain a high-quality native structure given an amino-acid sequence for possibly a large number of proteins [4].

Yet, in a largely detached thread in computational molecular biology, various researchers have advanced theory, experiment, and methods to reveal significant additional protein complexity; that is, proteins as inherently dynamic systems using often large motions to switch between different structures with which to bind to different molecular partners in the cell [5]. The dynamic view of proteins was evident in the early experimental structures obtained via Nuclear Magnetic Resonance (NMR); however, NMR is limited to reveal small structural fluctuations. Then came cryo-electron microscopy, which revealed the diversity of native tertiary structures assumed by a protein molecule [6].

Many researchers over the years have argued for broadening our computational treatment of proteins to account for the multiplicity of native structures [7]. However, the problem presents outstanding challenges, as it necessitates exploring a vast, high-dimensional space in search of possibly a very large number of functionally-relevant structures. The majority of computational methods that can reveal possibly multiple structures for a given protein leverage deep insight about a specific protein of interest. For instance, a line of work leverages experimentally-known structures of a protein to reveal latent coordinates over which to generate more structures [8], [9], [10], [11], [12], [13]. Other work is strictly limited to generating structures that mediate the transition between two given structures [14], [15], [16], [17], [18], [19], [20]. Several methods leverage collective coordinates to expedite numerical simulation (such as Molecular Dynamics simulation) [21], [22]. While beyond the scope of this work, adaptations to the classic Molecular Dynamics continue to be pursued to reveal the motion between two given structures or enhance the exploration of the structure space even when starting from a known structure rather than just the amino-acid sequence [23]. To the best of our knowledge, there are no *de-novo* methods that, given an amino-acid sequence alone, can reveal various functionally-relevant tertiary structures available to a protein.

We present such a method here. To handle a vast and high-dimensional search space, the method implements

• *A. B. Zaman, T. T. Inan, K. De Jong, and A. Shehu are with the Department of Computer Science, George Mason University, Fairfax, VA, 22030. A. Shehu is the corresponding author.*
E-mail: azaman6@gmu.edu, tinan@gmu.edu, kdejong@gmu.edu, amarda@gmu.edu

Manuscript received xxxxx xx, xxxx

adaptive stochastic search/optimization under the umbrella of evolutionary computation (EC) and leverages evolutionary search techniques to balance between exploration and exploitation in the presence of a finite computational budget. While EAs are naturally better suited to address the exploration-exploitation balance for complex optimization problems, existing EAs do not explicitly control this balance. We do so here by adaptively tuning the EA selection pressure to attain a proper balance between exploration and exploitation.

An earlier version of this method appeared recently in [24], where we demonstrated on benchmark and CASP-drawn datasets that the method managed to explore the energy minimum containing the known native structure of a given amino-acid sequence. In this paper, we extend the methodology and evaluate it on it's ability to identify multiple structures using a benchmark dataset we have constructed and present here for researchers to further advance work on this problem.

The rest of this paper proceeds as follows. A brief summary of related work and preliminaries are provided in Section 2. The methodology is presented in detail in Section 3 and evaluated in Section 4. Findings are further placed in context in Section 5, which concludes the paper.

2 Related Work and Preliminaries

The method we propose in this paper builds on at least two decades of work in computational biology and several contributions by our laboratory. In this section, we summarize the key developments that allow the reader to obtain a complete picture of the method. Where relevant, we point to other works that provide additional information not directly related to the presentation of the method in Section 3.

At a high-level, the presented method makes use of three components: (1) a way to represent a three-dimensional structure that allows for generating more structures; (2) a scoring/energy function that allows for comparing structures, determining which ones are more relevant for function/activity, and so connect between computation and theory; (3) an algorithm that utilizes (1) and (2) and so explores the structure space of a given amino-acid sequence by computing/sampling new structures in search of near-optima of the scoring function evaluated over computed/sampled structures.

2.1 The Structure Representation

The choice of how to represent a tertiary protein structure is key. In general, this choice has an oversized impact on the performance of an optimization algorithm and is studied in great detail in the EC community for various optimization problems [25]. For computations of protein structures, researchers have used a variety of representations, ranging from the straightforward Cartesian Coordinates, to idealized coordinates, dihedral/torsion angles, contact maps, distance matrices, etc. The representation employed in this paper is the dihedral angle-based representation, which is also popular with the popular Rosetta framework [26], Quark [2], and others. One of the reasons is expediency; given dihedral angles, forward kinematics can be used to directly obtain the Cartesian coordinates of the atoms; scoring functions operate over distances in Euclidean space. In contrast, contact maps and distance matrices need to be converted to tertiary structures. The process is not one-to-one and requires an additional optimization method (utilizing the contacts or distances as restraints). The other reason is related to the ability to utilize fragment to compute new structures from existing ones.

2.1.1 Molecular Fragment Replacement

A leap was made more than two decades ago, with the introduction of the molecular fragment replacement [27]. The realization by Baker and colleagues was that despite the diversity of protein structures, there is only a finite number of structural pieces/fragments utilized as legos to build tertiary structures. So, a simple approach was suggested. First, excise fixed-length pieces off known protein structures in databases, such as the Protein Data Bank (PDB) [28] to construct a fragment library indexed by fragment amino-acid sequences. Each fragment in the library was represented as a vector of backbone dihedral angles defined over a block $[i, j]$ of consecutive amino acids. This notation specifies that the fragment ranges from amino acid i and to amino acid j in the sequence. A fragment configuration for this fragment is comprised of a vector of $3{\cdot}(j{-}i{+}1)$ dihedral angles, with three dihedral angles specified for each amino acid (ϕ, ψ, and ω) that can be defined over bonds connecting consecutive backbone atoms in a fragment. The interested reader is referred to Ref. [29] for a review of protein geometry. Popular values for the chosen fragment length range in $\{3, 6, 9\}$. A new structure can be easily obtained by first choosing a random amino acid at position i in the chain and then replacing the configuration for fragment $[i, i + f - 1]$ in the current structure with a new configuration chosen from the fragment library. It is worth noting that, while the exposition above suggests one can easily construct their own fragment libraries, we elect instead to utilize those provided by Baker for a given amino-acid sequence, as the concept of fragments is extended to include sequence-similar ones (beyond identical amino acids).

2.1.2 Structure versus Conformation

We draw a distinction between structure and conformation. Specifically, a conformation is the result of instantiating variables selected to represent a tertiary structure. Since we utilize here the backbone dihedral angles to represent a protein tertiary structure, the selected variables are the backbone dihedral angles, and a conformation is a specific point in the variable space (of backbone dihedral angles). Through forward kinematics, we can go from conformation to structure (thus recovering the Cartesian coordinates of the atoms). The reverse is a straightforward calculation from coordinates to angles. It is worth noting that we do not carry these operations in house. The method we propose utilizes the Rosetta framework [26].

2.2 The Scoring Function

The conformation (and resulting structure) space available to a given amino-acid sequence is vast and high-dimensional. Some back-of-the-envelope calculations provide the context. Consider a short protein sequence of 60

amino acids. This results in 178 backbone dihedral angles (3 – ϕ, ψ, and ω – angles for each amino acid, save for the first two), thus giving rise to a 178-dimensional conformation space. Not all conformations correspond to energetically-favorable structures. Some conformations are clearly unfavorable, containing steric clashes among portions of the chain. Others are not energetically-favorable for a variety of reasons, captured in a scoring function that operates over Euclidean distances between atoms. The method we propose here utilizes the Rosetta `score4` scoring function that can operate over backbone atoms. Given a scoring function, the goal becomes to explore the conformation space in enough detail so as to reveal near-optimal structures. The lower the score, the more energetically-feasible a structure is; so, our method seeks to obtain a broad view of potentially different local minima of a given scoring function.

2.3 The Optimization Algorithm

It is not immediately clear how to devise an algorithm that can handle a vast, high-dimensional space and find many different local minima. Our research over the years has demonstrated that EAs have a higher exploration capability than gradient descent, Metropolis Monte Carlo (MMC), or even Simulated Annealing MMC (SA-MMC) [30], [31]. That is, given a finite computational budget, EAs see more of the structure space (or the associated scoring function). However, the measure of success of these EAs has been limited to getting to the local minimum housing a known native structure over benchmark test datasets popular for evaluating protein structure prediction methods that employ the single-structure view.

2.3.0.1 The EA framework: For the benefit of the reader, we summarize the basic ingredients of a hybrid EA (HEA) over which we build here. HEA is a population-based EA that evolves a fixed-size population over generations. The population consists of p conformations, which are generally referred to as individuals in EC literature. It is the task of the *initial population operator* to instantiate the first population of conformations. In each generation, the individuals in the current population are considered *parents* and *offspring* are produced from the parents via a *variation operator*. Specifically, the molecular fragment replacement technique described above is utilized in the variation operator. The offspring are then subjected to an *improvement operator* to further improve their (Rosetta) score, which is more generally referred to as *fitness* in EC literature. The employment of an improvement operator is what makes this EA a hybrid EA. The improved offspring then compete with the top parents, and a *selection operator* down-selects to p individuals that initialize the population for the next generation. Each of these operators are described in greater detail in related work [24], [32].

We have recognized from the body of earlier work that the enhanced exploration afforded by an EA has been key so as not to miss near-native structures in the presence of an inaccurate scoring function. The latter is an important point. All scoring functions, including the Rosetta suite, contain inherent biases and may not associate the lowest energy/score with a given native structure. This issue is exacerbated when the goal expands to providing a multiplicity of native structures, as the scoring function may be biased towards one or a portion of the structures, penalizing others that indeed are captured by experimental techniques.

2.3.1 Balancing between Exploration and Exploitation on the Fly

When the goal expands to obtaining diverse structures corresponding to different local minima in the scoring function, a proper balance between exploration (sampling more of the conformation space) and exploitation (drilling further down to find local minima) is critical. The EC setting exposes algorithmic knobs/parameters such as population size, variation, and selection, which can be varied to control the inherent trade-off between exploration and exploitation. Many EC researchers working on other optimization problems have focused on how to vary the values of these parameters in an adaptive manner [33], [34], [35], [36], [37].

While many approaches beyond the scope of this paper have been evaluated in EC literature, in this paper we pursue adaptive control. As we detail further in Section 3, the adaptation mechanism takes feedback from the optimization process, monitoring progress along selected statistics, and utilizes changes in these statistics to determine when and how to change parameter values. We note that adaptive parameter control is deemed to be the more effective way of adapting parameters, and so most research activities in EC literature are focused on it [38], [39].

Choosing what to adapt and when require careful consideration. Parameters that are generally adapted in EC literature are the variation operator, the population size, the representation, and/or the selection mechanism. Considering the two decades of research in modeling protein tertiary structures and our own body of work, we determine that the selection mechanism represents the most promising one for adaptation. In particular, in EAs, most of the exploitation comes from the applied selection mechanism. The decision when to adapt depends on detecting changes to statistics of interest. Two popular statistics in EC literature are fitness and diversity of the individuals (conformations in our case). Approaches that track fitness periodically check for change in best fitness of the individuals [40], the difference between the best fitness and the average fitness of a population [41], the fitness ranking of each individual [42], best-fitness frequency [43], or the success/fitness gain of the parameter values [44], [45]. Approaches that track diversity periodically check the Euclidean distances between individuals and the best-so-far individual [43], the Euclidean distances among individual solutions [46], the Hamming distance among individuals [47], or the diversity over the best, worst, and average fitness of individuals in the population [48]. The interested reader is referred to [38], [49] for a more detailed review of the feedback mechanisms used in EC literature. In this paper, we elect to use periodic change in best fitness.

3 METHODS

3.1 Underlying Framework: A Hybrid EA

As described in Section 2, to investigate the effects of changing selection pressure to obtain different exploration and exploitation capability in EAs, we build upon a baseline hybrid EA, the HEA proposed in [30] and evaluated against

Rosetta and others [30], [31]. HEA contains the basic evolutionary ingredients and evolves a fixed-size population of individuals/structures for a number of generations. For the selection mechanism, HEA uses truncation selection which is well-known to provide strong selection pressure that results in more exploitation and less exploration. From now, we refer to this baseline HEA algorithm as HEA-TR (TR for truncation). So as not to distract from the presentation, we provide more details on HEA-TR in the Appendix.

3.2 Main Ingredients of Adaptation Approach

As related in Section 2, we exert our control on the selection mechanism. So as to apply a wider range of selection pressure, we propose to switch between different selection schemes with different selection pressure to control the exploration-exploitation balance during an EA. This constitutes a novel approach and is the main methodological contribution of this paper. We believe this approach is of interest both more broadly for the EC and optimization community, as well as for computing diverse functionally-relevant structures of a protein.

The greediness (the propensity to select fitter individuals) of the selection mechanism is directly related to the exploration/exploitation pressure exerted by an EA. A greedier selection operator applies stronger selection pressure and results in more exploitation and less exploration of the search space. How to adjust the selection pressure on the fly during an EA is nontrivial and requires careful thinking. Adapting the tournament size parameter in tournament selection is appealing, but literature has shown that tuning this parameter in standard tournament selection to control exploration might lead to premature convergence [50]. Moreover, adjusting the tournament size to control selection pressure has its limits. At a lower bound (binary tournament), tournament selection applies much stronger selection pressure than a weak selection scheme, such as a uniform selection scheme [51]. On the other hand, weak selection pressure exerted by schemes such as uniform selection could be useful to prevent premature convergence and stagnation of the population.

Therefore, the selection schemes that we propose to use in this work are *uniform stochastic*, *fitness proportional*, *quaternary tournament*, and *truncation selection*. The reason for choosing these schemes is as follows. Uniform stochastic selection applies the weakest selection pressure. Truncation selection falls in the other end of the spectrum, exerting the strongest selection pressure. Fitness proportional selection applies stronger selection pressure than uniform selection. Although it provides less selection pressure than binary tournament selection, its exerted pressure is higher there is more diversity in the population [51]. Finally, the selection pressure exerted by quaternary tournament selection falls in between fitness proportional and truncation selection.

We first describe three variants of the HEA-TR algorithm, HEA-QT, HEA-FP, and HEA-US, depending on the selection mechanism employed, as we detail below. Finally, we describe HEA-AD, which implements the adaptive selection mechanism.

3.3 HEA-QT

In HEA-QT, the initial population, variation, and improvement operators described above are kept unchanged but the selection operator uses the quaternary tournament selection scheme instead of the truncation selection of HEA-TR. The idea is to reduce the selection pressure to decrease exploitation and promote exploration. In HEA-QT, all the parents and the improved offspring are first combined to form a selection pool S and each individual in S is evaluated using $score4$. Next, a 4-way tournament is held for each of the n spots in the population for the next generation, where n is size of the population. A uniform probability distribution is used to randomly pick 4 individuals from S with replacement and these 4 individuals then compete with each other to survive for the next generation. The fittest individual according to $score4$ wins the competition and is selected to fill the next open spot in the population for the next generation.

3.4 HEA-FP

HEA-FP employs the fitness proportional selection scheme instead of the truncation scheme of HEA-TR, while all other operators remain the same. Fitness proportional selection employs lesser selection pressure than quaternary tournament and truncation. In HEA-FP, all the parents and the improved offspring are combined to form a selection pool S. Then, each individual in S is assigned a selection probability proportional to their fitness. Specifically, an individual $x \in S$ is assigned a selection probability of $f(x)/\sum_{i \in S} f(i)$, where $f()$ measures the fitness of the individual according to $score4$. This distribution is then sampled n times to pick n individuals for the next generation (n is population size).

3.5 HEA-US

HEA-US applies the weakest selection pressure through uniform stochastic selection. As in HEA-QT and HEA-FP, all the other operators remain unchanged. A selection pool S of size $2n$ (n is the population size) is first formed which contains all the parents and the improved offspring. HEA-US assumes identical fitness for all the individuals; n individuals are picked from S uniformly at random to form the population for the next generation.

3.6 HEA-AD

In HEA-AD, we introduce an adaptive selection operator with the goal of achieve a better balance between exploration and exploitation. Instead of keeping the same selection pressure throughout the execution of the EA, HEA-AD adapts the selection pressure based on the characteristics of the evolving population. The algorithm periodically takes feedback from the population for a possible change of the selection pressure and raises or reduces the selection pressure accordingly.

HEA-AD keeps track of the *best-so-far fitness*, measured by the lowest $score4$ energy reached by any individual belonging to any population over all the generations up to the current generation c. We refer to this metric as $BSFF_c$. The reasons for choosing the $BSFF_c$ metric are as follows. $BSFF_c$ is simple and computationally efficient to compute.

In addition, a slowly improving $BSFF_c$ suggests the selection pressure is too weak; a strong selection pressure typically results in rapid improvements in $BSFF_c$ with a high risk of premature convergence [51]. Moreover, a lack of change in $BSFF_c$ for several generations can be an indicator that the population is stuck exploiting some parts of the space; in this case, a weaker selection pressure can be useful to achieve more exploration of the search space. So, in HEA-AD, every g generations, the adaptive mechanism compares the current *best-so-far fitness*, $BSFF_c$, to the best-so-far fitness observed over the g generations before. We refer to the latter as $BSFF_{c-g}$.

HEA-AD selects a selection scheme from the scheme pool SP = {uniform stochastic (US), fitness proportional (FP), quaternary tournament (QT), truncation (TR)}, sorted in increasing order of selection pressure, whenever a change of selection pressure is needed. The adaptive operator starts with a weaker selection scheme, FP, to encourage more exploration in early generations. In every g generations, the choice of selection scheme is revisited in the following way.

- If $BSFF_c$ increases by a small amount of $< s\%$ over $BSFF_{c-g}$, which suggests the selection pressure is too weak, the selection pressure is increased by replacing the current selection scheme with the next selection scheme in the pool SP which applies more selection pressure as the selection schemes are ordered from weakest to strongest in SP. For example, if the algorithm is using FP selection scheme at this point, it will go on to use QT from now.
- If $BSFF_c$ increases by a considerable amount of $> t\%$ over $BSFF_{c-g}$, which indicates too much exploitation is happening and the algorithm could be at risk of premature convergence, the selection pressure is increased by replacing the current selection scheme with the previous selection scheme in SP which applies less selection pressure. For example, if the current selection scheme in HEA-AD is the TR selection scheme, HEA-AD will then set the current selection scheme to be QT.
- If $BSFF_c$ is identical to $BSFF_{c-g}$ and the algorithm is currently using TR selection, the population could be stagnated and more exploration can help. So, the algorithm then keeps decreasing the selection pressure gradually by choosing the previous scheme in SP until the $BSFF_c$ improves.
- If $BSFF_c$ is identical to $BSFF_{c-g}$ and the algorithm is currently using US selection, this is an indication that the selection pressure kept decreasing from TR to end up in US. This suggests some exploration has already been performed by permitting weaker individuals to be selected and allowing them to produce offspring. So, more exploitation at this point could improve $BSFF_c$. Therefore, the algorithm then keeps increasing the selection pressure gradually for more exploitation by choosing the next scheme in SP until the $BSFF_c$ improves.

In HEA-AD, the above adaptive selection operator is employed to select individuals for the next generation. As in HEA-US, HEA-FP, and HEA-QT, the other operators (initial population, variation, and improvement) do not change. Rosetta $score4$ is used to measure the fitness of an individual.

3.7 Implementation Details

In all the EAs described above, the population size is $n = 100$ and the elitism rate for elitism-based truncation selection is $r = 25\%$, as in [30]. As is commonly done for conformation sampling (and EAs more generally), the termination criterion is set to the exhaustion of a fixed budget of fitness/energy evaluations. Specifically, the algorithms presented above are executed for a fixed budget of $10,000,000$ energy evaluations. This results in typically $120 - 300$K conformations sampled over $700 - 1600$ generations. For HEA-AD, the checking parameter g is set to 15; the change parameter s is set to 5, and t is set to 15. We note that no specific effort has been made to fine-tune these parameters to the problem at hand. All algorithms are implemented in Python and interface with the PyRosetta library. Each algorithm takes $1 - 3$ hours on one Intel Xeon E5-2670 CPU with 2.6GHz base processing speed and 20GB of RAM. The runtime range is mainly due to the different lengths of the amino-acid sequences of the target proteins. As we describe further in Section 4, the algorithms are run 5 times on each target protein's amino-acid sequence to account for possible variance.

4 Results

4.1 Experimental Setup

Our evaluation is organized along two major sets of experiments. In the first, the focus is on the single-structure prediction problem in order to carry out an ablation study and pitch against one another HEA-US, HEA-FP, HEA-QT, HEA-TR, and HEA-AD. We include Rosetta here, as it provides a baseline. We will refer to this as the *monomorphic* experimental setting. This evaluation shows HEA-AD to be superior according to several metrics. In the second set of experiments, we focus on the multiplicity of structures, to which we will refer as the *metamorphic* experimental setting from now on. We present a benchmark dataset collected over many research articles. Using this dataset, we evaluate HEA-AD over several metrics, comparing it to Rosetta as a baseline method and a recently-published EA that has been designed specifically with structure diversity in mind but not demonstrated in the metamorphic setting.

Each algorithm is run 5 times on each target to account for the stochasticity of the algorithms. We report the combined best performance over the 5 runs. Each run exhausts a fixed computational budget of $10,000,000$ energy evaluations for a total of $50,000,000$ energy evaluations for the 5 runs. Rosetta is run for $54,000,000$ energy evaluations on each target to conduct a fair comparison; each run of Rosetta exhausts $36,000$ energy evaluations and the total budget results in $1,500$ structures over $1,500$ runs.

As is practice in EAs for structure sampling [52], performance is measured on lowest reached energy and the lowest reached distance to the known native structure of the target. The former is important to analyze the effect of the selection mechanism on the exploration-exploitation tradeoff, and the latter is important as lower energies do not necessarily correlate with proximity to the native structure. We employ three popular proximity measure to calculate the distance between the sampled structure and the native structure;

root-mean-squared-deviation (RMSD) [53], Template Modeling Score (TM-Score) [54], and Global Distance Test - Total Score (GDT_TS) [55]. While RMSD is a dissimilarity metric (lower values correspond to better proximity), TM-score and GDT_TS are similarity metrics (higher values mean better proximity); the latter two provide a score in $[0, 1]$, whereas RMSD can be more affected by the chain length (number of amino acids). We report the GDT_TS score in percentage as is done in CASP competitions. Additionally, the comparison focuses on the main carbon atoms or the CA atoms of each amino acid as in CASP competitions. Finally, to provide a complete picture and measure how much better or worse performance is achieved on each target, we also employ performance profiles [56]. Performance profiles show the cumulative distribution functions for different performance ratios for a evaluation metric that reveal major performance characteristics.

To present a principled evaluation, we further strengthen our comparison with statistical significance tests. We utilize Fisher's [57] and Barnard's [58] exact tests for this purpose. Although Fisher's conditional test is widely adopted for statistical significance, Barnard's unconditional exact test is generally considered more powerful than Fisher's test for 2x2 contingency matrices.

4.2 Evaluation in the Monomorphic Setting

Recent work in [24] focused on the ablation study and limited itself to the monomorphic setting. The evaluation was carried over two datasets, a benchmark one introduced in [59] and enriched later with more targets [30], [60], [61], [62], consisting of 20 monomorphic proteins of different lengths and folds, and another one consisting of 10 targets drawn from hard, free-modeling targets from CASP12 and CASP13 competitions.

In the interest of space, we refrain from describing these datasets in detail, as such information has been presented in [24]. In addition, we do not repeat all the experiments related in [24] that establish the ability of HEA-AD to reach lower energy regions and to approach the single native structure closer than Rosetta and the other HEA variants over most of the targets; we also do not report on the statistical significance analysis that allows the evaluation in [24] to conclude that HEA-AD is superior in the monomorphic setting. Instead, we present an additional evaluation, effectively enriching that analysis, via performance profiles, on the benchmark dataset. We note that the analysis in [24] evaluates all the HEA variants and Rosetta against the submitted structure by the top ten groups over each of the CASP targets.

Let us briefly summarize the concept of performance profiles, as they have never been employed in protein modeling research to the best of our knowledge. Performance profiles provide us with a way of depicting how frequently a particular algorithm is within some distance of the best algorithm for a particular problem instance/target. So, for each problem instance, we first compute the best method, and then for every other method, we determine how far they are from optimal. In our case, problem instances are our targets in the dataset. We consider two separate metrics here, energy and RMSD. We vary the *performance ratio* over a range, limited $\in \{1.0, 3.0\}$ in our analysis. Specifically, for a given pr, *measure reached* means that an algorithm comes within a factor of pr of the best measure over all algorithms on a given target. The number of targets where an algorithm does this is tallied up, and this becomes indicative of its performance, also referred to as number of problems *solved*, at a given performance ratio.

Specifically, Figure 1(a) shows the performance profiles of each algorithm over the benchmark dataset of 20 targets in terms of the lowest energy reached. Figure 1(a) shows that the probability of HEA-AD to be the optimal algorithm among these 6 algorithms is about 0.55, considerably more than any of the other algorithms. At $pr = 1.2$, HEA-AD succeeds for 85% targets. HEA-QT reaches a success of 100% at a $pr = 1.38$, while HEA-AD and HEA-TR do so at $pr = 1.45$. Rosetta's performance profile rises very slowly and reaches 100% at $pr = 3.0$. Figure 1(b) relates a similar analysis focusing on the lowest RMSD to the native structure and shows that the probability of HEA-AD to be the optimal algorithm among these 6 algorithms is about 0.6, considerably more than any of the other algorithms. At $pr = 1.3$, HEA-AD succeeds for 85% targets. HEA-QT reaches a success of 100% at a $pr = 1.8$, while HEA-AD and HEA-FP do so at $pr = 2.0$. Rosetta saturates at $pr = 2.0$ with a success for 95% targets. These results clearly establish HEA-AD as the superior algorithm, enhancing the preliminary analysis in [24].

4.3 Evaluation on Metamorphic Dataset

In this setting, we focus on HEA-AD, shown superior over the other HEA variants by the above analysis. We include Rosetta as a baseline algorithm to understand how well a method exclusively designed for the monomorphic setting can perform on in the metamorphic setting. We also include another recent EA proposed by us in [63], SP-EA$^+$. SP-EA$^+$ aims to prevent premature convergence and retain diversity during optimization by evolving and maintaining multiple sub-populations, which is a popular construct in EC for optimization over multi-modal fitness landscapes.

We construct a novel dataset of 13 proteins, which we have compiled from various works [15], [64] and we detail below. The dataset consists mostly of proteins with two known native structures. Specifically, Table 1 relates the dataset. The first 12 rows relate proteins where wet-laboratories have elucidated two very distinct structures. The pairwise RMSDs are related in Column 4. The last row relates Calmodulin, for which 4 distinct structures are obtained from the PDB. The range of pairwise RMSD is shown in this case.

We first present a comparison of the three algorithms, Rosetta, SP-EA$^+$, and HEA-AD on the lowest-energy reached on the amino-acid sequence of each of the 13 proteins. As Table 2 shows, HEA-AD achieves the lowest energy on $9/13$ of the proteins in the dataset; Rosetta does so on $2/13$ cases, and SP-EA$^+$ on $2/13$ cases. HEA-AD comfortably outperforms Rosetta (10 vs. 3 cases) and SP-EA$^+$ (9 vs. 4 cases) in a head-to-head comparison. Panel (a) of Table 6 presents the p-values for statistical significance tests. These tests suggest that the performance improvements of HEA-AD in terms of lowest energy are statistically

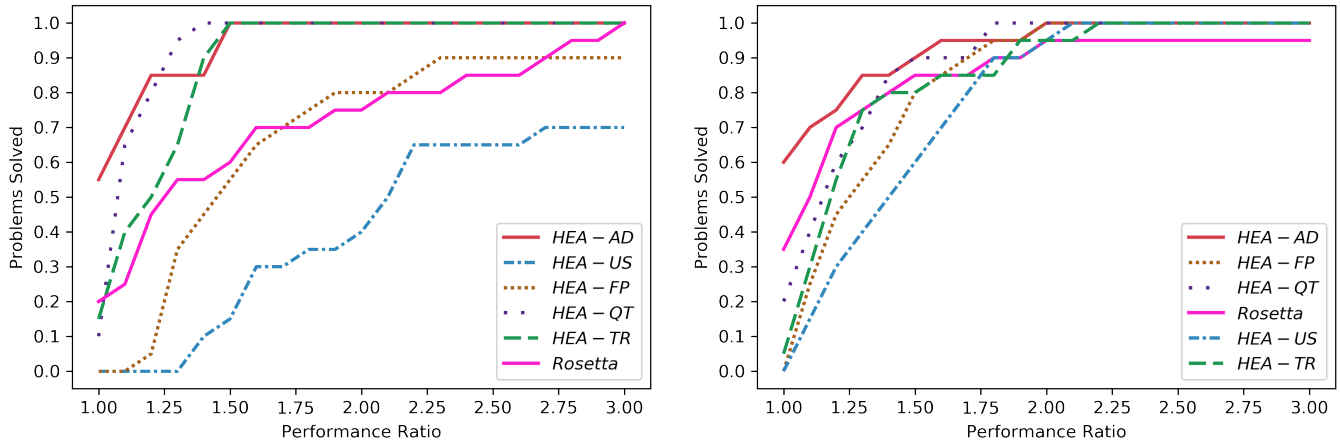


Fig. 1: Performance profiles for the algorithms on (a) lowest energy and (b) lowest RMSD metrics.

TABLE 1: Thirteen proteins with 2 or more known structures captured in the wet laboratory. Protein names are related in Column 1. Column 2 lists the length (number of amino acids) of each protein. Column 3 lists the PDB ids of the known structures, with the chain shown in parentheses.

Protein Name	Length	PDB Ids of Known Structures	RMSD(Å)
SARA	127	1fzp(D), 2frh(A)	19
Calcium-bound EF-Hand protein	134	1jfk(A), 2nxq(B)	15.9
Yeast Matalpha2/MCM1	87	1mnm(C), 1mnm(D)	6.7
IscA	112	1x0g(A), 1x0g(B)	18
NF-kB RelB	110	1zk9(A), 3jv6(A)	16.5
Beta 2 Microglobulin	100	3low(A), 3m1b(F)	19.3
Protein Related to DAN and Cerberus (PRDC)	148	4jph(B), 5hk5(H)	6.9
Methanocaldococcus jannaschii monomeric selecase	110	4qhf(A), 4qhh(A)	12.2
CopK	74	2k0q(A), 2lel(A)	9.1
SLAS-micelle bound alpha-synuclein	140	2kkw(A), 2n0a(D)	36.1
Human prion protein mutant HuPrP	147	2lej(A), 2lv1(A)	18.6
Cyanovirin-N	101	2ezm(A), 1l5e(A)	16
Calmodulin	148	1cfd(A), 1cll(A), 2f3y(A), 1lin(A)	4.3-13.4

significant at the 95% confidence level (p-values < 0.05) over both algorithms.

TABLE 2: Comparison of the lowest energy in Rosetta Energy Units (REUs) obtained by each algorithm under comparison on each of the 13 distinct proteins. The lowest energy value reached is marked in bold.

Lowest Energy (REU)		
Rosetta	SP-EA$^+$	HEA-AD
-111.8	-100.6	**-126.1**
-85.6	-87	**-97.9**
-71.3	-74.9	**-98**
-76	-73.8	-64.6
-52.7	**-56.4**	-53.1
-78	-84.5	**-104.3**
-44.2	-44.6	**-50**
-147.6	-138.1	**-155.8**
-136.2	-132.2	-125.3
-161.8	-164.1	**-169.7**
-124.4	-127.3	**-130.6**
-108.5	**-108.7**	-103
-200.7	-214.7	**-222**

Figure 2 shows the performance profiles of each of the three algorithms in terms of lowest energy. Figure 2 shows that the HEA-AD is the optimal algorithm on 0.7 of the proteins. HEA-AD "solves" all targets at a $pr = 1.2$, whereas SP-EA$^+$ and Rosetta do so at pr values of 1.32, and 1.37, respectively.

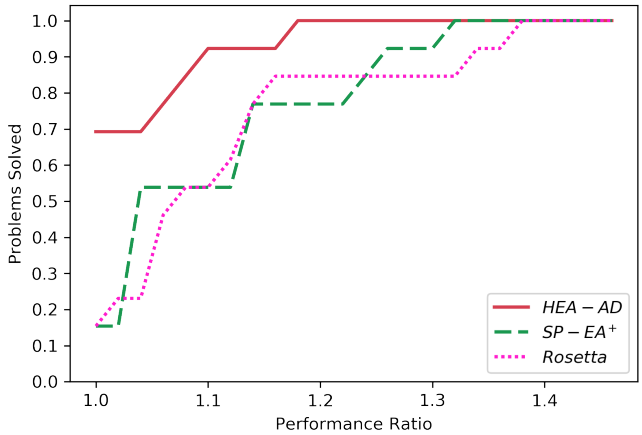


Fig. 2: Performance profiles for the algorithms on lowest energy on the metamorphic dataset.

The rest of the analysis now focuses on evaluating how close each algorithm comes to each of the listed structures for each target. We measure distance via RMSD, TM-Score, and GDT_TS. We expand the list of 13 proteins into 18 test cases, where we list the known structures as Target 1 and Target 2. This organization facilitates the exposition of our analysis. For Calmodulin, where 4 structures have been collected, this results in 6 Target 1 – Target 2 pairs: 1cfda –

1clla, 1cfda – 2f3ya, 1clla – f3ya, 1cfda – 1lina, 1clla – 1lina, and 2f3ya – 1lina.

Table 3 lists for each algorithm the lowest RMSD (over all conformations sampled by an algorithm over 5 runs) to each of the targets. Table 3 shows that for Target 1, HEA-AD achieves the lowest RMSD on $12/18$ cases, Rosetta in $2/18$ cases, and SP-EA$^+$ in $4/18$ cases. HEA-AD comfortably outperforms Rosetta (15 vs. 3 cases) and SP-EA$^+$ (13 vs. 6 cases) in a head-to-head comparison. Panel (b) in Table 6 shows that these performance improvements are statistically significant. For Target 2, HEA-AD achieves the lowest RMSD on $15/18$ cases, Rosetta in $1/18$ cases, and SP-EA$^+$ in $4/18$ cases. In a head-to-head comparison, HEA-AD easily outperforms Rosetta (17 vs. 2 cases) and SP-EA$^+$ (16 vs. 4 cases). Table 6(c) shows that these performance improvements are statistically significant.

TABLE 3: Comparison of the lowest RMSD obtained by each algorithm on each of the 18 target pairs. The lowest RMSD value reached is marked in bold.

Lowest RMSD (Å)					
Rosetta		SP-EA$^+$		HEA-AD	
Target 1	Target 2	Target 1	Target 2	Target 1	Target 2
8.5	7	6.6	6.1	**6.5**	**5.4**
2.2	10.2	**1.9**	1.5	2	**1.3**
5.6	7.8	4	3.3	**2.8**	**2.9**
6.1	12.4	6.4	9.2	6.4	**6.9**
11.2	9.3	8.5	**8.5**	**7.6**	9.3
12.1	**6.2**	8.8	7	**6.4**	6.9
12.3	11.9	9.9	**9.4**	**8.7**	**9.4**
6.6	10	6.7	7.9	**5.5**	**7.4**
4	7.6	**3.8**	4.4	4.1	**3.8**
10.1	31.2	9.2	13.6	**7.3**	**12.6**
9.1	17.6	9.9	**12.4**	10.5	**12.4**
6.6	9.1	6.8	6.5	**5.6**	**5.6**
6.8	8.3	3.2	**2.7**	**3**	2.8
6.8	3.8	3.2	3.6	**3**	**3.2**
8.3	3.8	**2.7**	3.6	2.8	**3.2**
6.8	4.3	3.2	3.7	**3**	**3.4**
8.3	4.3	**2.7**	3.7	2.8	**3.4**
3.8	4.3	3.6	3.7	**3.2**	**3.4**

Figure 3(a) and 3(b) show the performance profiles of each algorithm in comparison for the lowest RMSD metric for Target 1 and Target 2 in the metamorphic dataset respectively. Figure 3(a) shows that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the lowest RMSD to Target 1) is about 0.66, considerably more than the other algorithms. HEA-AD "reaches" Target 1 on all cases at $pr = 1.2$, whereas SP-EA$^+$ and Rosetta do so at pr values of 1.4 and 3.1, respectively. Figure 3(b) shows that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the lowest RMSD to Target 2) is about 0.83, considerably more than the other algorithms. HEA-AD "reaches" Target 2 on all cases at $pr = 1.15$, whereas SP-EA$^+$ does so at $pr = 1.3$; in contrast, Rosetta never reaches Target 2 on all cases in this pr range, saturating at 0.9 of the cases at $pr = 3.0$.

Tables 4 and 5 present the comparison in terms of TM-score and GDT_TS score (higher is better). Table 4 shows that for Target 1, HEA-AD achieves the highest TM-score on $15/18$ cases, Rosetta in $4/18$ cases, and SP-EA$^+$ in $1/18$ cases. In a head-to-head comparison, HEA-AD comfortably outperforms Rosetta (15 vs. 4 cases) and SP-EA$^+$ (16 vs. 2 cases) in a head-to-head comparison. Panel (d) in Table 6 shows that these performance improvements are statistically significant. For Target 2, HEA-AD achieves the highest TM-score on $15/18$ cases, Rosetta in $3/18$ cases, and SP-EA$^+$ in $2/18$ cases. In a head-to-head comparison, HEA-AD easily outperforms Rosetta (16 vs. 3 cases) and SP-EA$^+$ (17 vs. 2 cases). Panel (e) in Table 6 shows that these performance improvements are statistically significant.

Similarly, Table 5 shows that for Target 1, HEA-AD achieves the highest GDT_TS on $12/18$ cases, Rosetta in $4/18$ cases, and SP-EA$^+$ in $3/18$ cases. In a head-to-head comparison, HEA-AD comfortably outperforms Rosetta (15 vs. 4 cases) and SP-EA$^+$ (13 vs. 5 cases) in a head-to-head comparison. Panel (f) in Table 6 shows that these performance improvements are statistically significant. For Target 2, HEA-AD achieves the highest TM-score on $15/18$ cases, Rosetta in $2/18$ cases, and SP-EA$^+$ in $1/18$ cases. In a head-to-head comparison, HEA-AD easily outperforms Rosetta (16 vs. 2 cases) and SP-EA$^+$ (17 vs. 1 cases). Panel (g) in Table 6 shows that these performance improvements are statistically significant.

TABLE 4: Comparison of the highest TM-score obtained by each algorithm on each of the 18 target pairs. The highest TM-score value reached is marked in bold.

Highest TM-score					
Rosetta		SP-EA$^+$		HEA-AD	
Target 1	Target 2	Target 1	Target 2	Target 1	Target 2
0.33	0.46	0.41	0.48	**0.42**	**0.55**
0.77	0.46	0.73	0.84	0.72	**0.87**
0.58	0.62	0.67	0.66	**0.7**	**0.69**
0.5	**0.44**	0.41	0.39	0.45	**0.44**
0.36	**0.38**	0.34	0.32	**0.36**	0.34
0.32	**0.5**	0.41	0.46	**0.44**	0.47
0.29	0.28	0.31	**0.33**	**0.33**	0.32
0.45	0.34	0.44	0.42	**0.55**	**0.45**
0.66	0.51	**0.66**	0.55	0.64	**0.59**
0.29	0.15	0.38	0.21	**0.48**	**0.23**
0.36	0.29	0.37	**0.4**	**0.4**	**0.4**
0.47	0.35	0.44	0.49	**0.48**	**0.58**
0.48	0.48	0.73	0.76	**0.74**	**0.81**
0.48	0.69	0.73	0.69	**0.74**	**0.77**
0.48	0.69	0.76	0.69	**0.81**	**0.77**
0.48	0.62	0.73	0.71	**0.74**	**0.72**
0.48	0.62	0.76	0.71	**0.81**	**0.72**
0.69	0.62	0.69	0.71	**0.77**	**0.72**

Figure 4(a) and 4(b) show the performance profiles of each algorithm in terms of the highest TM-score metric for Target 1 and Target 2 on the metamorphic dataset, respectively. Figure 4(a) shows that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the highest TM-score to Target 1) is about 0.83, which is considerably more than the other algorithms. HEA-AD "reaches" Target 1 on all cases at $pr = 1.15$, whereas SP-EA$^+$ and Rosetta do so at pr values of 1.3 and 1.7, respectively. Figure 4(b) shows that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the highest TM-score to Target 2) is about 0.83, which is again considerably more than the other algorithms. HEA-AD "reaches" Target 2 on all cases at $pr = 1.15$, whereas SP-EA$^+$ and Rosetta do so at $pr = 1.2$ and $pr = 1.9$, respectively.

Figure 5(a) and 5(b) show the performance profiles of each algorithm in terms of the highest GDT_TS metric for Target 1 and Target 2 on the metamorphic dataset, respec-

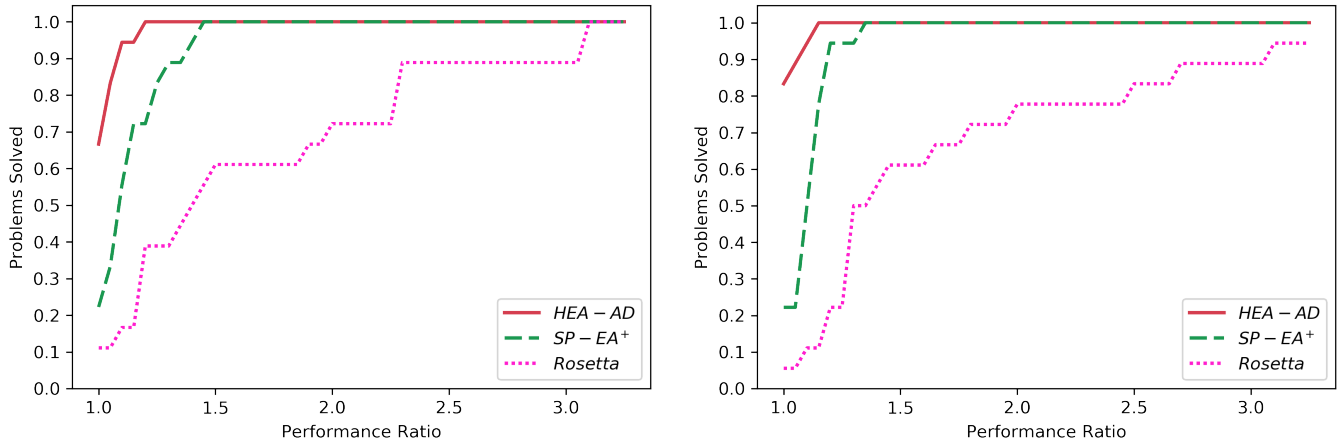


Fig. 3: Performance profiles for the algorithms on lowest RMSD for (a) Target 1 and (b) Target 2 on the metamorphic dataset.

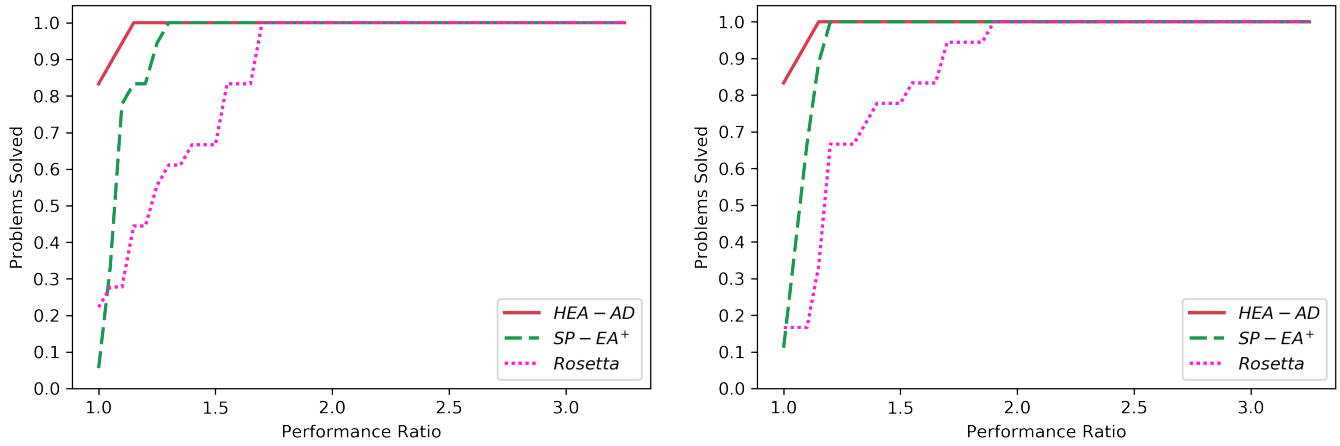


Fig. 4: Performance profiles for the algorithms on highest TM-score for (a) Target 1 and (b) Target 2 on the metamorphic dataset.

TABLE 5: Comparison of the highest GDT_TS score obtained by each algorithm on each of the 18 target pairs. The highest GDT_TS value reached is marked in bold.

Highest GDT_TS (%)					
Rosetta		SP-EA$^+$		HEA-AD	
Target 1	Target 2	Target 1	Target 2	Target 1	Target 2
34.5	43	39.75	47.25	**41.5**	**53.25**
81.25	50.78	80.86	88.28	77.34	**89.84**
59.96	62.99	69.08	70.13	**75**	**71.1**
47	40.5	43.75	38	45.25	**41.5**
33.66	**35.4**	34.16	31.19	**34.65**	33.17
31.67	**48.74**	39.68	43.45	**42.68**	44.19
26.63	27.53	28.05	**30.18**	**28.83**	28.81
42.82	33.35	44.77	41.9	**50.93**	**44.91**
67.91	52.39	66.55	59.8	**67.91**	**62.5**
26.79	12.68	32.36	15.61	**40.89**	**16.43**
29.42	21.25	31.12	32.14	**34.52**	**33.33**
52.48	39.36	46.29	46.53	46.04	**52.72**
43.06	40.8	**61.02**	67.19	60.24	**70.83**
43.06	60.42	**61.02**	58.51	60.24	**65.1**
40.8	60.42	67.19	58.51	**70.83**	**65.1**
43.06	54.34	**61.02**	60.94	60.24	**62.33**
40.8	54.34	67.19	60.94	**70.83**	**62.33**
60.42	54.34	58.51	60.94	**65.1**	**62.33**

tively. Figure 5(a) shows that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the highest GDT_TS to Target 1) is about 0.66, which is higher than the other algorithms. HEA-AD "reaches" Target 1 on all cases at $pr = 1.15$, whereas SP-EA$^+$ and Rosetta do so at pr values of 1.3 and 1.75, respectively. Figure 5(b) shows that the probability of HEA-AD to be the optimal algorithm among all (in terms of reaching the highest GDT_TS to Target 2) is about 0.61, again higher than the other algorithms. At $pr = 1.1$, HEA-AD succeeds on 95% targets. HEA-AD "reaches" Target 2 on all cases at $pr = 2$, whereas SP-EA$^+$ and Rosetta do so at $pr = 2.1$ and $pr = 2.6$, respectively.

4.3.1 *Visualization of Structures*

The quality of the structures obtained by HEA-AD is shown qualitatively in Fig. 6, which draws from the structures obtained by HEA-AD that are closest to 4 distinct structures of Calmodulin (PDB ids 1cfda, 1clla, 2f3ya, and 1lina, respectively). Fig. 6 shows that HEA-AD captures each of these structures reasonably well (with RMSDs shown for each).

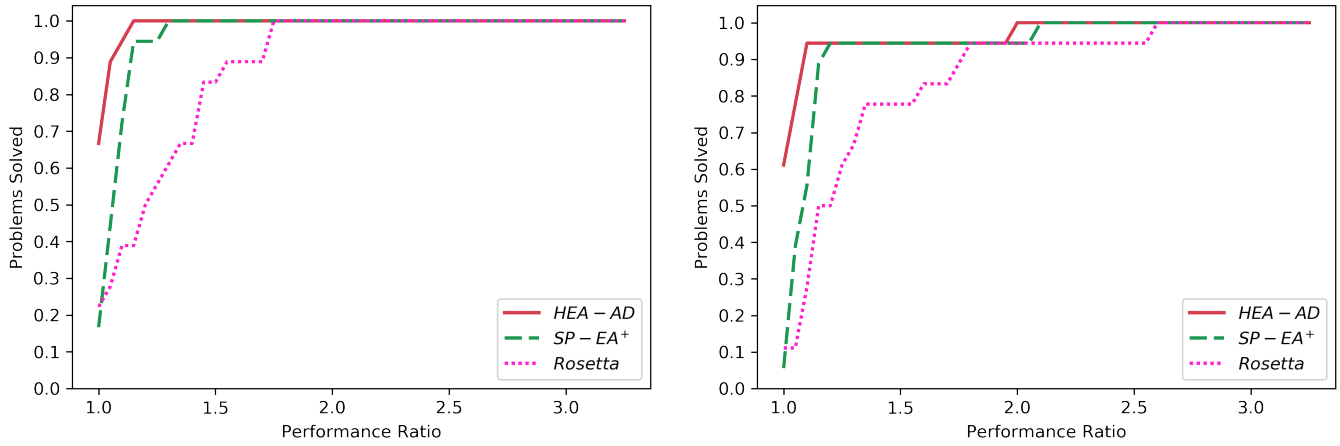


Fig. 5: Performance profiles for the algorithms on highest GDT_TS score for (a) Target 1 and (b) Target 2 on the metamorphic dataset.

TABLE 6: Results for the 1-sided Fisher's and Barnard's tests on the comparisons presented in Table 2, 3, 4, and 5 on the metamorphic dataset. The tests evaluate the null hypothesis that HEA-AD does not achieve (a) lower lowest energy, (b) lower lowest RMSD on Target 1, (c) lower lowest RMSD on Target 2, (d) higher highest TM-score on Target 1, (e) higher highest TM-score on Target 2, (f) higher highest GDT_TS score on Target 1, (g) higher highest GDT_TS score on Target 2 in comparison to a particular algorithm. p-values less than 0.05 are marked in bold.

Test	Rosetta	SP-EA$^+$
(a) Fisher's	**0.008466**	0.05762
Barnard's	**0.004729**	**0.03778**
(b) Fisher's	**7.60E-05**	**0.02186**
Barnard's	**3.48E-05**	**0.01443**
(c) Fisher's	**3.22E-07**	**6.61E-05**
Barnard's	**1.14E-07**	**2.51E-05**
(d) Fisher's	**3.05E-04**	**2.62E-06**
Barnard's	**1.58E-04**	**9.71E-07**
(e) Fisher's	**1.48E-05**	**3.22E-07**
Barnard's	**6.46E-06**	**1.14E-07**
(f) Fisher's	**3.05E-04**	**0.009197**
Barnard's	**1.58E-04**	**0.00569**
(g) Fisher's	**2.62E-06**	**3.58E-08**
Barnard's	**9.71E-07**	**9.71E-09**

5 Conclusion

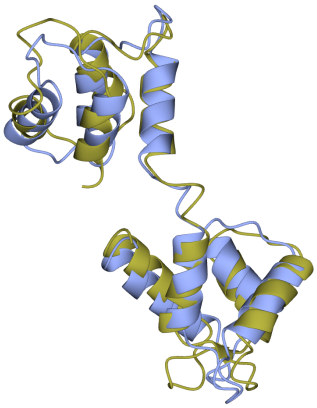
The results presented above show that the adaptive selection mechanism in HEA-AD balances the exploitation and exploration effectively and samples regions of the structure space that contain better-scoring structures. Analysis over diverse metrics establishes the superiority of HEA-AD not only over other HEA variants, but also Rosetta and other EAs. In particular, the evaluation in the metamorphic setting, for which we construct a dataset that we hope will be adopted and enriched to serve as a benchmark dataset, shows that HEA-AD is superior and can capture diverse structures several angstroms away when only utilizing the amino-acid sequence of a given protein (and no other structural information about the protein at hand). Further analysis related in the Appendix relates that even in the presence of strong bias in the Rosetta score4 function towards some structures (and against others), HEA-AD still manages to capture the various structures known for a protein. This is another indication of the high exploration power of HEA-AD, further suggesting its ability to take into account the multiplicity of native structures. We believe that these results warrant further research on more powerful stochastic optimization algorithms. In particular, we point to growing work on generative deep learning. The majority of these methods are not yet able to condition to a given amino-acid sequence and mostly relate the ability to generate protein-like tertiary structures in the sequence-agnostic setting. Other deep learning frameworks still consider the narrow setting of one single structure, leveraging high-inductive bias. We believe that the integration of deep learning models and EAs presents opportunities to further make inroads into what still remains a challenging problem.

Acknowledgments

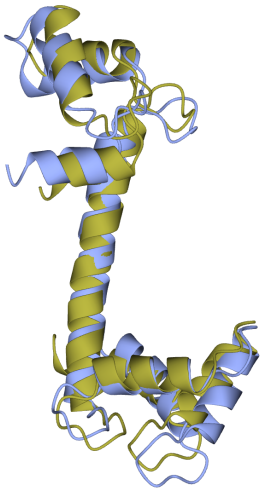
This work is supported in part by the National Science Foundation under Grant No. 1900061. Computations were run on ARGO, a research computing cluster provided by the Office of Research Computing at George Mason University, VA (URL: http://orc.gmu.edu). This material is additionally based upon work by AS supported by (while serving at) the National Science Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We thank Dr. Nasrin Akhter and Prof. Alexander Brodsky for useful discussions and feedback on this work.

References

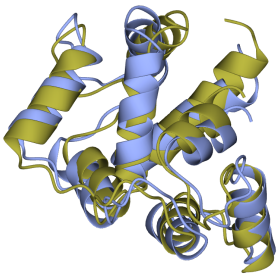
[1] D. D. Boehr and P. E. Wright, "How do proteins interact?" *Science*, vol. 320, no. 5882, pp. 1429–1430, 2008.
[2] H. Deng, J. Y., , and Y. Zhang, "Protein structure prediction," *Int J Mod Phys B*, vol. 32, no. 18, p. 1840009, 2018.
[3] J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, and A. Tramontano, "Critical assessment of methods of protein structure prediction (casp)—round x," *Proteins: Structure, Function, and Bioinformatics*, vol. 82, pp. 1–6, 2014.



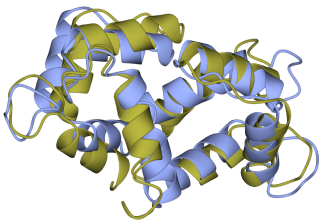
1cfda (3Å)



1clla (2.8Å)



2f3ya (3.2Å)



1lina (3.4Å)

Fig. 6: The HEA-AD structure closest to the known Calmodulin structures under PDB ID 1cfda, 1clla, 2f3ya, and 1lina is drawn in blue; the wet-laboratory structures are drawn in olive. Rendering is performed with the CCP4mg molecular graphics software [65].

[4] E. Callaway, "'it will change everything': DeepMind's AI makes gigantic leap in solving protein structures," 2020.

[5] D. D. Boehr, R. Nussinov, and P. E. Wright, "The role of dynamic conformational ensembles in biomolecular recognition," *Nature Chem Biol*, vol. 5, no. 11, pp. 789–96, 2009.

[6] E. Callaway, "The revolution will not be crystallized," *Nature*, vol. 525, pp. 172–174, 2015.

[7] K. Jenzler-Wildman and D. Kern, "Dynamic personalities of proteins," *Nature*, vol. 450, pp. 964–972, 2007.

[8] R. Clausen, B. Ma, R. Nussinov, and A. Shehu, "Mapping the conformation space of wildtype and mutant H-Ras with a memetic, cellular, and multiscale evolutionary algorithm," *PLoS Comput Biol*, vol. 11, no. 9, p. e1004470, 2015.

[9] R. Clausen and A. Shehu, "A data-driven evolutionary algorithm for mapping multi-basin protein energy landscapes," *J Comp Biol*, vol. 22, no. 9, pp. 844–860, 2015.

[10] E. Sapin, D. B. Carr, K. A. De Jong, and A. Shehu, "Computing energy landscape maps and structural excursions of proteins," *BMC Genomics*, vol. 17, no. Suppl 4, p. 456, 2016.

[11] E. Sapin, K. A. De Jong, and A. Shehu, "From optimization to mapping: An evolutionary algorithm for protein energy landscapes," *IEEE/ACM Trans Comput Biol and Bioinf*, 2017, doi: 10.1109/TCBB.2016.2628745.

[12] T. Maximova, E. Plaku, and A. Shehu, "Structure-guided protein transition modeling with a probabilistic roadmap algorithm," *IEEE/ACM Trans Comput Biol and Bioinf*, 2017, doi: 10.1109/TCBB.2016.2586044.

[13] T. Maximova, Z. Zhao, D. B. Carr, E. Plaku, and A. Shehu, "Sample-based models of protein energy landscapes and slow structural rearrangements," *J Comput Biol*, vol. 25, no. 1, pp. 33–50, 2017.

[14] K. Molloy, R. Clausen, and A. Shehu, "A stochastic roadmap method to model protein structural transitions," *Robotica*, vol. 34, no. 8, pp. 1705–1733, 2016.

[15] K. Molloy and A. Shehu, "A general, adaptive, roadmap-based algorithm for protein motion computation," *IEEE Trans. NanoBioSci.*, vol. 2, no. 15, pp. 158–165, 2016.

[16] P. Koehl, "Minimum action transition paths connecting minima on an energy surface," *J Chem Phys*, vol. 18, no. 145, p. 184111, 2016.

[17] D. Devaurs, K. Molloy, M. Vaisset, and A. Shehu, "Characterizing energy landscapes of peptides using a combination of stochastic algorithms," *IEEE Trans. NanoBioSci.*, vol. 14, no. 5, pp. 545–552, 2015.

[18] K. Molloy and A. Shehu, "Elucidating the ensemble of functionally-relevant transitions in protein systems with a robotics-inspired method," *BMC Struct. Biol.*, vol. 13, no. Suppl 1, p. S8, 2013.

[19] N. Haspel, M. Moll, M. L. Baker, W. Chiu, and K. L. E., "Tracing conformational changes in proteins," *BMC Struct Biol*, vol. 10, no. Suppl1, p. S1, 2010.

[20] H. Huang, E. Ozkirimli, and C. B. Post, "A comparison of three perturbation molecular dynamics methods for modeling conformational transitions," *J Chem Theory Comput*, vol. 5, no. 5, pp. 1301–1314, 2009.

[21] M. Gur, J. D. Madura, and I. Bahar, "Global transitions of proteins explored by a multiscale hybrid methodology: Application to adenylate kinase," *Biophys J*, vol. 105, no. 1643–1652, p. 184111, 2013.

[22] M. Gur, E. Zomot, M. H. Cheng, and I. Bahar, "Energy landscape of leut from molecular simulations," *J Chem Phys*, vol. 143, p. 243134, 2015.

[23] B. J. Grant, A. A. Gorfe, and J. A. McCammon, "Ras conformational switching: Simulating nucleotide-dependent conformational transitions with accelerated molecular dynamics," *PLoS Comput Biol*, vol. 5, no. 3, p. e1000325, 2015.

[24] A. Zaman, T. Tinan, and A. Shehu, "Protein decoy generation via adaptive stochastic optimization for protein structure determination," in *IEEE Intl Conf on Bioinformatics and Biomedicine (BIBM)*, 2020.

[25] K. De Jong, *Parameter Setting in EAs: a 30 Year Perspective*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–18.

[26] A. Leaver-Fay, M. Tyka, S. M. Lewis, O. F. Lange, J. Thompson, R. Jacak *et al.*, "ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules," *Methods Enzymol*, vol. 487, pp. 545–574, 2011.

[27] K. T. Simons, C. Kooperberg, E. Huang, and D. Baker, "Unexpected features of the dark proteome," *J Mol Biol*, vol. 268, pp. 209–225, 1997.

[28] H. M. Berman, K. Henrick, and H. Nakamura, "Announcing the worldwide Protein Data Bank," *Nat. Struct. Biol.*, vol. 10, no. 12, pp. 980–980, 2003.

[29] A. Shehu, "Conformational search for the protein native state," in *Protein Structure Prediction: Method and Algorithms*, H. Rangwala and G. Karypis, Eds. Fairfax, VA: Wiley Book Series on Bioinformatics, 2010, ch. 21.

[30] B. Olson and A. Shehu, "Multi-objective stochastic search for sampling local minima in the protein energy surface," in *ACM Conf on Bioinf and Comp Biol (BCB)*, Washington, D. C., September 2013, pp. 430–439.

[31] ——, "Multi-objective optimization techniques for conformational sampling in template-free protein structure prediction," in *Intl Conf on Bioinf and Comp Biol (BICoB)*, Las Vegas, NV, 2014, pp. 143–148.

[32] B. Olson, K. A. De Jong, and A. Shehu, "Off-lattice protein structure prediction with homologous crossover," in *Conf on Genetic and Evolutionary Computation (GECCO)*. New York, NY: ACM, 2013, pp. 287–294.

[33] B. W. Goldman and D. R. Tauritz, "Meta-evolved empirical evidence of the effectiveness of dynamic parameters," in *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, ser. GECCO '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 155–156. [Online]. Available: https://doi.org/10.1145/2001858.2001945

[34] J. Hesser and R. Männer, "Towards an optimal mutation probability for genetic algorithms," in *Parallel Problem Solving from Nature*, H.-P. Schwefel and R. Männer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 23–32.

[35] J. Smith and T. C. Fogarty, "Self adaptation of mutation rates in a steady state genetic algorithm," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 318–323.

[36] T. Bäck, "The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm," in *Parallel Problem Solving from Nature*. Elsevier, 1992.

[37] J. Cervantes and C. R. Stephens, "Limitations of existing mutation rate heuristics and how a rank ga overcomes them," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 369–397, 2009.

[38] A. Aleti and I. Moser, "A systematic literature review of adaptive parameter control methods for evolutionary algorithms," *ACM Comput. Surv.*, vol. 49, no. 3, Oct. 2016.

[39] K. De Jong, *Parameter Setting in EAs: a 30 Year Perspective*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–18.

[40] A. E. Eiben, E. Marchiori, and V. A. Valkó, "Evolutionary algorithms with on-the-fly population size adjustment," in *Parallel Problem Solving from Nature - PPSN VIII*, X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiňo, A. Kabán, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 41–50.

[41] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.

[42] M. Sewell, J. Samarabandu, R. Rodrigo, and K. Mcisaac, "The rank-scaled mutation rate for genetic algorithms," *Information Technology - IT*, 01 2006.

[43] M. Giger, D. Keller, and P. Ermanni, "Aorcea - an adaptive operator rate controlled evolutionary algorithm," *Comput. Struct.*, vol. 85, no. 19–20, p. 1547–1561, Oct. 2007.

[44] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, ser. Problemata (Stuttgart). Frommann-Holzboog, 1973. [Online]. Available: https://books.google.com/books?id=-WAQAQAAMAAJ

[45] L. Davis (ed.), *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991.

[46] B. McGinley, J. Maher, C. O'Riordan, and F. Morgan, "Maintaining healthy population diversity using adaptive crossover, mutation, and selection," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 692–714, 2011.

[47] J. Maturana, Á. Fialho, F. Saubion, M. Schoenauer, F. Lardeux, and M. Sebag, *Adaptive Operator Selection and Management in Evolutionary Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 161–189.

[48] L. Budin, M. Golub, and D. Jakobovic, "Parallel adaptive genetic algorithm." 01 1998, pp. 157–163.

[49] A. E. Eiben and J. E. Smith, *Parameter Control in Evolutionary Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 129–151.

[50] H. Xie and M. Zhang, "Tuning selection pressure in tournament selection," in *Technical Report Series, School of Engineering and Computer Science*. Victoria University of Wellington, New Zealand, 2009.

[51] K. A. De Jong, *Evolutionary Computation: a Unified Approach*. Cambridge, MA: MIT Press, 2006.

[52] A. Shehu, "A review of evolutionary algorithms for computing functional conformations of protein molecules," in *Computer-Aided Drug Discovery*, ser. Methods in Pharmacology and Toxicology, W. Zhang, Ed. Springer Verlag, 2015.

[53] A. D. McLachlan, "A mathematical procedure for superimposing atomic coordinates of proteins," *Acta Cryst A*, vol. 26, no. 6, pp. 656–657, 1972.

[54] Y. Zhang and J. Skolnick, "Scoring function for automated assessment of protein structure template quality," *Proteins: Structure, Function, and Bioinformatics*, vol. 57, no. 4, pp. 702–710, 2004.

[55] A. Zemla, "Lga: a method for finding 3d similarities in protein structures," *Nucleic acids research*, vol. 31, no. 13, pp. 3370–3374, 2003.

[56] E. Dolan and J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, 03 2001.

[57] R. A. Fisher, "On the interpretation of χ^2 from contingency tables, and the calculation of P," *J Roy Stat Soc*, vol. 85, no. 1, pp. 87–94, 1922.

[58] G. A. Barnard, "A new test of 2x2 tables," *Nature*, vol. 156, p. 177, 1945.

[59] J. Meiler and D. Baker, "Coupled prediction of protein secondary and tertiary structure," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 21, pp. 12 105–12 110, 2003.

[60] G. Zhang, L. Ma, X. Wang, and X. Zhou, "Secondary structure and contact guided differential evolution for protein structure prediction," *IEEE/ACM Trans Comput Biol and Bioinf*, 2018, preprint.

[61] A. Zaman, P. Kamranfar, C. Domeniconi, and A. Shehu, "Reducing ensembles of protein tertiary structures generated de novo via clustering," *Molecules*, vol. 25, no. 9, p. 2228, 2020.

[62] A. Zaman and A. Shehu, "Building maps of protein structure spaces in template-free protein structure prediction," *Journal of Bioinformatics and Computational Biology*, vol. 17, no. 06, p. 1940013, 2019.

[63] A. Zaman, K. A. De Jong, and A. Shehu, "Using subpopulation eas to map molecular structure landscapes," in *Conf on Genetic and Evolutionary Computation (GECCO)*. New York, NY: ACM, 2019, pp. 1–8.

[64] N. Chen, M. Das, A. LiWang, and L.-P. Wang, "Sequence-based prediction of metamorphic behavior in proteins," *Biophysical Journal*, vol. 119, no. 7, pp. 1380–1390, 2020.

[65] S. McNicholas, E. Potterton, K. S. Wilson, and M. E. M. Noble, "Presenting your structures: the CCP4mg molecular-graphics software," *Acta Cryst*, vol. D76, pp. 386–394, 2011.

[66] A. Shmygelska and M. Levitt, "Generalized ensemble methods for de novo structure prediction," *Proc. Natl. Acad. Sci. USA*, vol. 106, no. 5, pp. 94 305–95 126, 2009.

Ahmed Bin Zaman Ahmed Bin Zaman is a Ph.D. candidate in the Department of Computer Science at George Mason University. He received his Master of Science degree in Computer Science from George Mason University in 2020 and his Bachelor of Science degree in Computer Science and Engineering from Shahjalal University of Science and Technology, Bangladesh, in 2012. He worked as a team leader, researcher, and developer at Technext Limited, Bangladesh, for three years before joining Metropolitan University, Bangladesh, as a lecturer in 2015. His research interests include evolutionary computation, artificial intelligence, and computational biology.

Toki Tahmid Inan Toki Tahmid Inan is a Ph.D. student in the Department of Computer Science at George Mason University. He received his Bachelor of Science in Computer Science and Engineering from Military Institute of Science and Technology (MIST), Bangladesh, in 2018. Before joining Mason, he was a research assistant at MIST and an analyst at Humans and Software, Bangladesh. His research interests include computational biology and artificial intelligence.

Kenneth De Jong Dr. Kenneth De Jong is a Professor of Computer Science and an Associate Director of the Krasnow Institute, George Mason University. His research interests include genetic algorithms, evolutionary computation, machine learning, and adaptive systems. He is currently involved in research projects involving the development of new evolutionary algorithm (EA) theory, the use of EAs as heuristics for NP-hard problems, and the application of EAs to the problem of learning task programs in domains such as robotics, diagnostics, navigation, and game playing. Support for these projects is provided by the US National Science Foundation, DARPA, ONR, and NRL. He is an active member of the Evolutionary Computation research community and has been involved in organizing many of the workshops and conferences in this area. He is the founding editor-in-chief of the journal Evolutionary Computation (MIT Press), and a member of the board of ACM SIGEVO. He is a member of the IEEE and the ACM.

Amarda Shehu Dr. Amarda Shehu is a Professor in the Department of Computer Science in the Volgenau School of Engineering. She is Co-Director of the Center for Advancing Human-Machine Partnerships (CAHMP), a Transdisciplinary Center for Advanced Study at George Mason University. Shehu obtained her Ph.D. from Rice University in 2008. Her research focuses on novel algorithms in artificial intelligence and machine learning to bridge between computer and information science, engineering, and the life sciences. Her laboratory has made many contributions in bioinformatics and computational biology regarding the relationship between macromolecular sequence, structure, dynamics, and function. Shehu has published over 150 technical papers with postdoctoral, graduate, undergraduate, and high-school students. She is currently the chair of the steering committee of the ACM/IEEE Journal on Transactions in Bioinformatics and Computational Biology, where she is also an associate editor. Shehu is the recipient of an NSF CAREER Award, and her research is regularly supported by various NSF programs, including Information Integration and Informatics, Robust Intelligence, Computing Core Foundations, and Software Infrastructure, as well as various state and private research awards.

APPENDIX

HEA-TR Algorithm

In HEA-TR, the initial population is obtained with great care by applying an initial population operator. From an amino-acid sequence of a target protein, the initial population operator first creates n identical extended chains, where n is the size of the population, in Rosetta's centroid representation. For each amino-acid, the representation only models the heavy-backbone atoms and a pseudo-atom representing the centroid of the side chain atoms. To randomize each of these n extended chains, a two-stage MMC search is utilized. The stages mainly differs in the fitness functions they use and the value of the acceptance probability scaling parameter of the Metropolis criterion. The goal for the first stage is to randomize the extended chains while avoiding steric clashes (self collisions). To do so, it employs Rosetta $score0$ energy function that encourages steric repulsion. The acceptance probability is set to 0 which ensures that only a move that decreases $score0$ is accepted. The second stage employs the $score1$ energy function which encourages the formation of secondary structures (α-helices and β-sheets). The scaling parameter for this stage is set to a higher value of 2 to which adds diversity of the individuals. Each move in this MMC search is a fragment replacement of length 9.

Each individual in the population is considered a parent and a variation operator is applied to a parent to produce an offspring. The variation operator applies a single fragment replacement of length 3 that introduces a small structural change over a parent. Applying fragment replacement on each of the n parents, we obtain n offspring, where n is the size of the population. Any offspring generated by the variation operator is subjected to an improvement operator that employs a local search to map the offspring to a nearby local minima in the energy surface. The local search is greedy in nature and only the moves that lower energy are accepted. Each move in the local search is a fragment replacement of length 3 and the improvement operator utilizes Rosetta $score3$ scoring function to evaluate the moves which encourages the formation of compact tertiary structures [66]. The search ends when l consecutive moves fail to decrease the energy of a structure according to $score3$, where l is the number of amino acids in the structure.

Offspring and parents compete for survival via the selection operator. HEA uses elitism-based truncation selection where each parent and improved offspring is first evaluated using Rosetta's full centroid scoring function $score4$ that considers short- and long-range hydrogen bonding in addition to the energetic terms in $score3$. Top $r\%$ individuals from the parents then compete for survival with the improved offspring, where r is the elitism rate. Since truncation selection is used in HEA, the competing individuals are sorted in increasing order of their fitness according to $score4$, and the fittest n individuals are selected to survive and form the next generation.

5.1 Evaluating Energy Function Bias

We evaluate here the bias that Rosetta has against specific structures over the 13 proteins with 2 or more structure captured in the wet laboratory. The following experiment is carried out. First, the Rosetta $score4$ function is used to evaluate the energy of each structure. These values are reported as *pre-relax* in Columns 3-4 in Table 7. Columns 5-6 then show the energy of each structure for Target 1 and Target 2, respectively, after applying Rosetta's $FastRelax$ protocol to fix any mispositions of atoms in the wet-laboratory structures. Columns 7-8 show the RMSD between the structures before and after $FastRelax$ for Target 1 and Target 2, respectively.

Table 7 explains why Rosetta does not perform as well as HEA-AD and SP-EA$^+$ (see manuscript). For example, for the pair {1jfk(A), 2nxq(B)}, Rosetta reaches within 2.2Å for 1jfk(A) but 10.2Å for 2nxq(B); similarly, for the pair {1cll(A), 2f3y(A)}, Rosetta reaches within 3.8Å for 2f3y(A) but 8.3Å for 1cll(A).

Table 7 provides a reason, as it shows that Rosetta has a bias towards specific structures. For example, for the pair {1jfk(A), 2nxq(B)}, the energy of 1jfk(A) (32.2 REU) before relaxation is higher than that of 2nxq(B) (16.7 REU) before relaxation; after relaxation, the energy for 1jfk(A) (-28.2 REU) is much lower than that of 2nxq(B) (-12 REU). This helps understand why Rosetta reproduces 1jfk(A) much better than 2nxq(B). Similarly, the energy for 2f3y(A) (-183.4 REU) before relaxation is much lower than that of 1cll(A) (-129 REU); this holds even after relaxation (-163.8 REUs for 2f3y(A) and -144.5 REUs for 1cll(A)). This helps understand why Rosetta reproduces 2f3y(A) much better than 1cll(A). Similar observations can be drawn on the other cases. However, unlike Rosetta, even though using a Rosetta scoring function, HEA-AD captures the different structures much better than Rosetta. This is another indication of the high exploration power of HEA-AD that allows it to take into account the multiplicity of native structures.

TABLE 7: Energy of native structure for each metamorphic target pre- and post- $FastRelax$ is shown in Columns 3-6. RMSD between the native structures before and after $FastRelax$ is shown in Columns 7-8. The PDB IDs of the metamorphs are shown in Columns 1-2.

Target 1	Target 2	Target 1 Energy Pre-relax	Target 2 Energy Pre-relax	Target 1 Energy Post-relax	Target 2 Energy Post-relax	RMSD between Target 1 Pre-relax & Post-relax	RMSD between Target 2 Pre-relax & Post-relax
1fzpd	2frha	112.5	35.2	7	-21.4	5.3	1.8
1jfka	2nxqb	32.2	16.7	-28.2	-12	1.4	3.7
1mnmc	1mnmd	-22.1	-31.2	-46	-41.5	2.8	6.9
1x0ga	1x0gb	-4	88.5	-43.5	41.9	1.3	4
1zk9a	3jv6a	67.2	-12.3	12.7	-75.6	12.2	0.8
3lowa	3m1bf	77	10.8	57.2	-47.5	11.6	1.2
4jphb	5hk5h	6.4	88.1	10.4	62.6	0.8	4
4qhfa	4qhha	-125.1	-59.1	-118.5	-71.8	1.2	4.7
2k0qa	2lela	6.7	55.3	-39.3	-13.3	2.2	2.6
2kkwa	2n0ad	-2.1	315.5	0.3	202.4	24.9	22
2leja	2lv1a	49.2	52.7	1.1	41.8	5.8	9.2
2ezma	1l5ea	-55.3	15.4	-88.7	5.1	0.8	6.8
1cfda	1clla	-89.9	-129	-154.6	-144.5	3.6	3.3
1cfda	2f3ya	-89.9	-183.4	-154.6	-163.8	3.6	2.1
1clla	2f3ya	-129	-183.4	-144.5	-163.8	3.3	2.1
1cfda	1lina	-89.9	-164.4	-154.6	-150.9	3.6	3.3
1clla	1lina	-129	-164.4	-144.5	-150.9	3.3	3.3
2f3ya	1lina	-183.4	-164.4	-163.8	-150.9	2.1	3.3