Protein Decoy Generation via Adaptive Stochastic Optimization for Protein Structure Determination

Ahmed Bin Zaman
Dept. of Computer Science
George Mason University
Fairfax, VA, USA
azaman6@gmu.edu

Toki Tahmid Inan
Dept. of Computer Science
George Mason University
Fairfax, VA, USA
tinan@gmu.edu

Amarda Shehu
Dept. of Computer Science
George Mason University
Fairfax, VA, USA
amarda@gmu.edu

2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 978-1-7281-6215-7/20/$31.00 ©2020 IEEE DOI: 10.1109/BIBM49941.2020.9313102

***Abstract*—Many regions of the protein universe remain inaccessible by wet-laboratory or homology modeling methods. Elucidating these regions necessitates structure determination *in silico*. Protein structure determination in the absence of a structural template remains a challenging task with two core problems, known as decoy generation and decoy selection. In this paper, we address the problem of decoy generation, which inherently involves exploring the unknown, vast, and high-dimensional structure space of a given amino-acid sequence in the presence of a finite computational budget for relevant structures. Leveraging a stochastic optimization framework, we first demonstrate how selection pressure can be employed to control the trade-off between exploration and exploitation. Moreover, we then propose a novel algorithm that tunes its behavior towards exploration or exploitation as needed via an adaptive selection mechanism. We present a thorough evaluation on 30 protein targets in a comparative setting, where we compare the proposed adaptive algorithm to state-of-the-art algorithms that include the top ten groups in the two recent CASP competitions. The results show that the proposed algorithm is not only competitive against several of these groups, but it additionally outperforms several of them on many targets, suggesting that adaptive stochastic optimization is a promising framework for decoy generation.**

***Keywords*-protein structure prediction, decoy generation, stochastic optimization, tertiary structure**

I. Introduction

Protein structure determination remains a hallmark problem in molecular biology. First, the recognition is due to the knowledge that the three-dimensional/tertiary structure determines to a great extent the biological activities in which a protein molecule participates in the living cell [1]. Second, the problem poses outstanding challenges in wet and dry laboratories; currently, many regions of the protein universe remain inaccessible. A recent study estimates that 44–54% of the proteome in eukaryotes and viruses ($\sim 546,000$ proteins) and 14% of the proteome in archaea and bacteria is dark [2].

Computational progress is closely tracked biannually via the "Critical Assessment of protein Structure Prediction" (CASP) community-wide experiments [3]. A prominent category assesses the ability to predict the biologically-active/native tertiary structure of a given protein amino-acid sequence for which there is no structural template from a close or remote homolog (with sufficiently similar sequence). The methods that operate in the absence of a template are known as ab initio or template-free protein structure determination/prediction, or more generally as free modeling.

Free modeling is a task with two inherent problems, *decoy generation* and *decoy selection.* The term decoy is used to denote a computed tertiary structure to convey that a set of computed decoys may *hide* within them the actual near-native structures. The objective in decoy selection is to tease out the relevant/near-native structures from the generated decoys.

Decoy generation methods operate under an optimization framework, whether one considers the popular Rosetta platform [4], Quark [5], or the much-publicized AlphaFold [6] in the latest CASP. At their core, these methods seek to optimize a scoring function. The hypothesis is that the native structure resides at the global minimum. The structure space is vast and high-dimensional, all current scoring functions are inherently inaccurate, and the surface of the scoring function is rugged; in response, the assumption is relaxed, and the objective becomes seeking local minima of the scoring function that are populated by near-native structures [7].

A leap was made more than two decades ago, with the introduction of fragment replacement for decoy generation [8]. The realization was that though the protein structure universe is large, there are only a finite number of different structural pieces/fragments if one "cut" known protein structures into fixed-length fragments. Simple Monte Carlo-based algorithms were debuted, which at every Monte Carlo step proposed replacing the structure of a fragment selected at random over the backbone chain to vary the current decoy and obtain a new one. A Metropolis criterion based on score improvement provided the bias towards local minima.

Many optimization algorithms followed, including many EAs by our group over the years [9]–[11]. Other groups focus on improving the accuracy of scoring functions and/or devising novel fragments [12]. The recent AlphaFold falls in this latter category, as well, learning a scoring function, generating novel fragments, and then carrying out a gradient descent over the learned scoring function, putting together decoys with fragments [6]. Other groups avoid tertiary structure and fragments and operate instead over other representations, such as contact maps or distance maps. Deep learning is emerging as powerful in this thread of research [13]. It is worth noting

This work is supported in part by NSF FET Grant No. 1900061.

that optimization is again needed to go from contact or distance maps to the actual tertiary structures [14].

In this paper, we address the problem of decoy generation. Our research over the years has demonstrated that casting an optimization algorithm under the umbrella of evolutionary computation results in powerful evolutionary algorithms (EAs) with higher exploration capability than gradient descent, Metropolis Monte Carlo (MMC), or even Simulated Annealing MMC (SA-MMC) [9]–[11], [15]. Given a finite computational budget, these algorithms see more of the structure space (or the associated scoring function). This is key to obtaining diverse but physically-realistic decoys so as not to miss the near-native ones in the presence of an inaccurate scoring function.

The evolutionary computation setting exposes algorithmic knobs that can be varied to control the inherent trade-off between exploration (seeing more of the space) and exploitation (getting to better-scoring regions of the space). In this paper, we focus on how to better control this trade-off. We first demonstrate how selection pressure is useful for this purpose. Learning from our observations, we then propose a novel adaptive algorithm that tunes its behavior towards exploration or exploitation as needed via an adaptive selection mechanism.

In our experiments, we consider 30 protein targets that include recent hard CASP targets from the free modeling category. We generate dozens of thousands of decoys for each target, and analyze their quality in comparison to results by the top ten groups in the two recent CASP competitions. The results, presented in Section III, show that the proposed algorithm is competitive and even outperforms on many targets, suggesting that adaptive stochastic optimization is a promising framework for decoy generation. We place our findings in context in Section IV, which concludes the paper. Before proceeding with methodological details in Section II, we provide a brief summary of related work in Section I-A.

A. Related Work and Preliminaries

Let us first provide some more detail behind the concept of a fragment and a fragment configuration, as these are key concepts utilized in this paper. Given a chain of n amino acids, numbered from the N- to the C-terminus, we specify a fragment $[i, j]$. This notation indicates that the fragment starts at amino acid i and ends at amino acid j in the sequence $[1, \ldots, n]$. A configuration for this fragment consists of a vector of $2 \cdot (j - i + 1)$ dihedral angles, with two dihedral angles, ϕ and ψ, specified for each amino acid. The reader is referred to Ref. [16] for a review of protein geometry.

Known native structures of proteins in the Protein Data Bank (PDB) [17] can be excised into fixed-length fragments of some chosen length f. The resulting fragment configurations are organized in a fragment configuration library indexed by fragment amino-acid sequences. A new decoy can be easily obtained by varying an existing decoy at a selected (often at random) amino acid i; the configuration for the fragment $[i, i + f - 1]$ in the decoy is replaced with a new configuration for a same or similar-sequence fragment chosen from a fragment configuration library.

Fragment replacement is a key unit in many decoy generation platforms. Lengths employed in Rosetta are 9 and 3. Quark makes use of longer fragments [18]. AlphaFold [6] augments fragment configuration libraries with novel fragments generated from a generative recurrent neural network.

Different methods use different scoring functions and different optimization algorithms. Rosetta and Quark use SA-MMC. AlpaFold uses a simple gradient descent. Others use single-objective or multi-objective EAs to improve upon simulated annealing [9]–[11], [14], [19]. While EAs are naturally better equipped at addressing the balance between exploration and exploitation for complex optimization problems, the existing algorithms do not explicitly control this balance. We do so here by managing the EA selection pressure to achieve a proper exploration-exploitation balance.

In this paper, we build over a hybrid evolutionary algorithm (HEA) [9], which we describe briefly; the interested reader can find further details in Ref. [9]). HEA is a population-based EA that evolves a fixed-size population of p individuals (decoys) over generations. The *initial population operator* instantiates the first population. In each generation, the individuals in the population are considered *parents* and *offspring* are produced from the parents via a *variation operator*. The offspring are then subjected to an *improvement operator* to further improve their score, which is more generally referred to as *fitness*. 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.

Given the amino-acid sequence of a protein target, the initial population operator first constructs p identical *extended* chains using Rosetta's centroid representation [4], which, for each amino-acid, models only the heavy-backbone atoms and a pseudo-atom representing the centroid of the side chain atoms. The chains are then randomized via a two-stage MMC search where each move is a fragment replacement of length 9. The first stage is greedy and employs the Rosetta $score0$ scoring function that encourages steric repulsion. The second stage employs the Rosetta $score1$ to encourage the formation of secondary structures and uses the Metropolis criterion.

Each individual in the population is subjected to a variation operator to obtain an offspring, which applies a single fragment (of length 3) replacement to a parent decoy. Each offspring is then subjected to an improvement operator. It uses a greedy local search to map the offspring to a nearby local minimum in the Rosetta $score3$ energy function landscape. Its moves are fragments replacements over fragments of length 3. The search ends when k consecutive moves fail to decrease the score, where k is the length of the target sequence.

HEA uses a elitism-based truncation selection operator. Essentially, all individuals (parents and improved offspring) are first evaluated using Rosetta's full centroid scoring function $score4$. The top-scoring $l\%$ individuals from the parents are combined with the improved offspring to compete for survival; l is the elitism rate. These individuals are then sorted in increasing order of their $score4$, and the top p individuals are selected to represent the population for the next generation.

II. METHODS

The selection mechanism is a very powerful knob via which one can control the balance between exploration and exploitation and directly control the quality of generated decoys. For this reason, we first design three variants of the HEA algorithm summarized in Section I-A. These variants only change the selection mechanism, utilizing the other operators (initial population, variation, and improvement) as in HEA.

To make prominent the fact that HEA uses truncation-based selection, we refer to it as HEA-TR from now on. We note that it is well-understood that truncation-based selection places very strong selection pressure over a population and tips the scales towards more exploitation and less exploration. In the three variants we design, we gradually weaken the selection pressure, thus tipping the scales towards more exploration than exploitation, to observe its impact over the quality of generated decoys. We name these three variants HEA-QT, HEA-FP, and HEA-US and describe them in greater detail below.

Finally, we propose a fourth algorithm, HEA-AD, to achieve an appropriate balance between exploration and exploitation. HEA-AD tunes its behavior towards exploration or exploitation as needed via an adaptive selection mechanism. We now provide further details.

A. The HEA-QT Algorithm

HEA-QT uses quaternary tournament as the selection mechanism, which applies weaker selection pressure than truncation selection. The goal is to decrease the selection pressure so as to reduce exploitation and promote exploration. In HEA-QT, the parents and the improved offspring are first evaluated via $score4$ and combined to construct a selection pool. Then, for each of the p "open spots" in the population for the next generation (p is population size), a 4-way tournament is held. 4 individuals are randomly selected from the selection pool using a uniform probability distribution with replacement. The top individual among the selected 4 according to $score4$ is designated winner and so selected to survive for the next generation (the selected individual fills the next open spot).

B. The HEA-FP Algorithm

HEA-FP employs a fitness-proportional selection scheme. Fitness proportional selection applies even less selection pressure than quaternary tournament. In HEA-FP, each individual i in the combined selection pool S of parents and improved offspring is assigned a selection probability of $fitness(i)/\sum_{j\in S} fitness(j)$, where $fitness()$ is the Rosetta $score4$ of an individual. We sample this distribution p times to select p individuals to survive for the next generation, where p is the size of the population.

C. The HEA-US Algorithm

HEA-US applies the weakest selection pressure through uniform stochastic. Uniform stochastic selection assumes identical fitness for all the individuals. In HEA-US, the parents and the improved offspring are combined to form a selection pool of size $2p$ (p is the population size). p individuals are then selected from it uniformly at random for the next generation.

D. HEA-AD Algorithm

HEA-AD employs an adaptive selection operator to better balance between exploration and exploitation. Instead of keeping the same selection pressure over generations, HEA-AD adapts the selection pressure based on the characteristics of the population. The algorithm evaluates the population every few generations for possible adjustments in the selection pressure and decreases or increases the selection pressure as needed.

Specifically, the adaptive mechanism periodically checks for a possible change of the selection pressure. The algorithm tracks the *best-so-far fitness*, which measures the best fitness (lowest Rosetta $score4$) over the g populations over the past g generations. Let us refer to this statistic as BSFF.

When a change needs to be made, as detailed below, the algorithm chooses a new selection mechanism from a scheme pool SP = {uniform stochastic, fitness proportional, quaternary tournament, truncation}. The pool is sorted in ascending order of selection pressure. HEA-AD first starts with a weaker selection scheme, the fitness proportional one, so as to encourage more exploration in the early generations. Over every g generations, the choice of the selection scheme is revisited as follows.

If the current BSFF (over the last g generations) increases by $< 5\%$ over the BSFF observed over g generations earlier, the selection pressure is increased by replacing the current selection scheme with the next one in the pool SP that applies more selection pressure. Recall that the selection schemes are ordered from weakest to strongest. For example, if the current selection mechanism in HEA-AD is the fitness proportional one, HEA-AD will then set the current selection mechanism to be quaternary tournament. A slowly rising best-so-far curve suggests the selection pressure is too weak [20].

If the current BSFF (over the last g generations) increases by $> 15\%$ over the BSFF observed over g generations earlier, we take this as indication that too much exploitation is happening. The algorithm can converge prematurely to a suboptimal minimum. Therefore, the selection pressure is decreased by switching the current selection scheme with the previous scheme in SP. For example, if the algorithm is using truncation selection at this point, it will go on to use quaternary tournament from now on.

If the the current BSFF (over the last g generations) is unchanged from the BSFF observed over g generations earlier, and the algorithm is currently using truncation selection, the population could be stagnated and more exploration can help. Therefore, the selection pressure is decreased gradually by choosing the previous scheme in SP until the BSFF improves.

If the current BSFF (over the last g generations) is unchanged from the BSFF observed over g generations earlier, and the algorithm is currently using uniform stochastic selection, this indicates that the selection pressure has kept decreasing from truncation to end up in uniform stochastic. So, some exploration has already been performed by selecting weaker individuals and allowing them to reproduce. Therefore, we can now aim to improve the BSFF. The selection pressure

is increased gradually for more exploitation by choosing the next scheme in SP until the BSFF improves.

This adaptive selection operator is utilized in HEA-AD algorithm to select individuals for the next generation. As with the other variants, the initial population, variation, and improvement operators remain unchanged and the fitness of an individual is evaluated via the Rosetta $score4$.

E. Implementation Details

The population size is $p = 100$; the elitism rate for elitism-based truncation selection is set to $l = 25\%$, as in [9]. As is common for decoy generation, the termination criterion is set to a total budget of fitness/score evaluations, 10M here. This results in typically $120 - 300$K decoys generated over $700 - 1600$ generations. The checking parameter g for HEA-AD is set to 15. The algorithms are implemented in Python and interface with the PyRosetta library. The algorithms take $1 - 3$ hours on one Intel Xeon E5-2670 CPU with 2.6GHz base processing speed and 20GB of RAM. The runtime differs mainly due to different lengths of the target proteins. The algorithms are run 5 times on each target protein's amino-acid sequence to account for the effects of stochasticity.

III. Results

A. Experimental Setup

We evaluate on two datasets. The first is a benchmark dataset, first introduced in [21] and enriched with more targets [9], [22], [23]. The dataset consists of 20 target proteins of varying lengths (53 to 146 amino acids) and folds (α, β, $\alpha + \beta$, and $coil$). The second dataset is also used in recent literature [10], [14], [24], [25] and contains 10 hard, free-modeling targets from CASP12 and CASP13.

We first compare HEA-TR, HEA-QT, HEA-FP, HEA-US, and HEA-AD to one another and Rosetta's decoy generation algorithm on the benchmark dataset. We then compare HEA-AD's performance on the CASP targets dataset against the top 10 performing groups on each target, as listed on the CASP website (https://predictioncenter.org/casp13/index.cgi).

Our algorithms are run 5 times on each target to count for the stochastic optimization; the best performance over 5 runs combined is reported. Each run exhausts $10M$ energy evaluations. For a fair comparison, Rosetta is run for $54M$ energy evaluations on each target. Rosetta is evaluation expensive, and one run of it exhausts 36K score evaluations. The above total budget results in $1,500$ decoys over $1,500$ runs.

Performance is measured on best-achieved score and best proximity to a known native structure. The score-based analysis is particularly important to expose the impact of the selection mechanism on the exploration-exploitation trade-off [26]. We make use of two popular metrics to measure proximity of decoys to the native structure; root-mean-squared-deviation (RMSD) [27] and Global Distance Test - Total Score (GDT_TS) [28]. RMSD is a dissimilarity metric (lower values indicate better proximity), GDT_TS is a similarity metric (higher values indicate better proximity). GDT_TS is reported in CASP competitions as a percentage value, as we do here. In addition, as is practice in CASP, the comparison focuses on the CA atoms (the main carbon atom of each amino acid).

Finally, we carry out statistical significance tests for a principled evaluation. We utilize Fisher's [29] and Barnard's [30] 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.

B. Evaluation on Benchmark Dataset

Figure 1 shows the best score (lowest Rosetta $score4$) over decoys generated by HEA-TR, HEA-QT, HEA-FP, HEA-US, HEA-AD, and Rosetta for each benchmark target. In the interest of clarity, a target is named by the entry id of a representative native structure known for it in the PDB. Figure 1 shows that HEA-AD achieves the best score on $11/20$ of the target proteins. In comparison, Rosetta achieves the best score in $4/20$, HEA-TR in $3/20$, and HEA-QT in $2/20$ of the targets. In a head-to-head comparison, HEA-AD comfortably outperforms each of the other algorithms. P-values for the statistical significance tests are listed in Table I. The p-values indicate that the performance (score) improvements of HEA-AD over other algorithms are statistically significant at the 95% confidence level (p-values < 0.05).

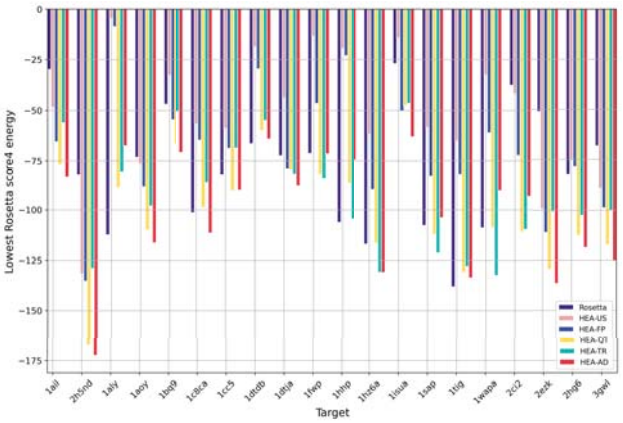

Fig. 1. Comparison of the best Rosetta $score4$ (measured in Rosetta Energy Units - REUs) over decoys generated by each of the algorithms on each target in the benchmark dataset. The algorithms are distinguished by color-coding.

More observations can be drawn from Figure 1 and Table I(b). HEA-QT outperforms all algorithms comfortably, except HEA-AD. Its performance improvements are statistically significant at the 95% confidence level. This confirms that truncation selection exerts strong selection pressure that results in premature convergence. As quaternary tournament applies less selection pressure, it is able to better explore more of the space. On the other hand, HEA-TR easily outperforms HEA-FP and HEA-US; Table I(c) indicates the performance improvements are statistically significant at the 95% confidence level. These results show that fitness proportional and uniform selection apply too little selection pressure, resulting in little exploitation.

Similar analysis is now presented on the best proximity to the known native structure reached over generated decoys for

TABLE I

RESULTS FOR THE 1-SIDED FISHER'S AND BARNARD'S TESTS ON HEAD-TO-HEAD COMPARISONS IN FIGURE 1. THE TESTS EVALUATE THE NULL HYPOTHESIS THAT (A) HEA-AD DOES NOT ACHIEVE, (B) HEA-QT DOES NOT ACHIEVE, (C) HEA-TR DOES NOT ACHIEVE BETTER SCORE ON THE BENCHMARK DATASET IN COMPARISON TO A PARTICULAR ALGORITHM; P-VALUES LESS THAN 0.05 ARE MARKED IN BOLD.

Test	Rosetta	HEA-TR	HEA-QT	HEA-FP	HEA-US
(a) HEA-AD					
Fisher's	**0.01282**	**0.01282**	0.05642	**7.25E-12**	**7.25E-12**
Barnard's	**0.008299**	**0.008299**	**0.04035**	**9.10E-13**	**9.10E-13**
(b) HEA-QT					
Fisher's	0.05642	**0.01282**	N/A	**2.91E-09**	**7.25E-12**
Barnard's	**0.04035**	**0.008299**	N/A	**7.47E-10**	**9.10E-13**
(c) HEA-TR					
Fisher's	N/A	N/A	N/A	**0.01282**	**2.91E-09**
Barnard's	N/A	N/A	N/A	**0.008299**	**7.47E-10**

TABLE II

RESULTS FOR THE 1-SIDED FISHER'S AND BARNARD'S TESTS ON HEAD-TO-HEAD COMPARISONS IN FIGURE 2. THE TESTS EVALUATE THE NULL HYPOTHESIS THAT (A) HEA-AD DOES NOT ACHIEVE, (B) HEA-QT DOES NOT ACHIEVE, (C) HEA-TR DOES NOT ACHIEVE LOWER LOWEST RMSD ON THE BENCHMARK DATASET IN COMPARISON TO A PARTICULAR ALGORITHM. P-VALUES LESS THAN 0.05 ARE MARKED IN BOLD.

Test	Rosetta	HEA-TR	HEA-QT	HEA-FP	HEA-US
(a) HEA-AD					
Fisher's	0.05642	**0.0006159**	**0.001528**	**1.52E-10**	**1.68E-09**
Barnard's	**0.04035**	**0.0003401**	**0.0006061**	**3.73E-11**	**3.83E-10**
(b) HEA-QT					
Fisher's	0.3762	0.05548	N/A	**1.11E-06**	**7.25E-12**
Barnard's	0.3179	**0.03517**	N/A	**2.97E-07**	**9.10E-13**
(c) HEA-TR					
Fisher's	N/A	N/A	N/A	0.5	**0.02808**
Barnard's	N/A	N/A	N/A	0.4373	**0.01924**

a target. Figure 2 shows the best RMSD (lowest) over decoys generated by HEA-TR, HEA-QT, HEA-FP, HEA-US, HEA-AD, and Rosetta for each benchmark target. Figure 2 shows that HEA-AD achieves the best RMSD on $12/20$ of the targets. In comparison, Rosetta achieves the best RMSD in $7/20$ of the targets, HEA-QT in $4/20$, and HEA-TR in $1/20$ of the targets. In a head-to-head comparison, HEA-AD comfortably outperforms each of the other algorithms. P-values for the statistical significance tests are listed in Table II. The p-values indicate that the performance (score) improvements of HEA-AD over other algorithms are statistically significant at the 95% confidence level (p-values < 0.05).

Additional observations can be drawn. For instance, HEA-QT outperforms all the other algorithms except HEA-AD. Table II(b) indicates that performance improvements over HEA-TR, HEA-FP, and HEA-US are statistically significant at the 95% confidence level. Moreover, HEA-TR outperforms HEA-FP and HEA-US. Table II(c) indicates that the performance improvement over HEA-US is statistically significant.

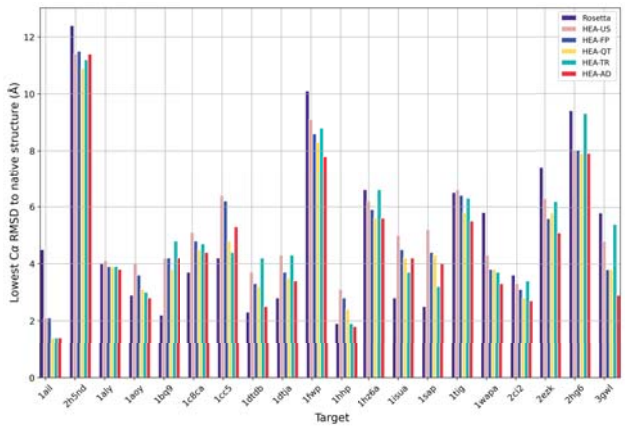

Fig. 2. Comparison of the best/lowest RMSD (measured in Å) to the native structure over decoys generated by each of the algorithms on each target in the benchmark dataset. The algorithms are distinguished by color-coding.

C. Evaluation on CASP Dataset

Since we do not have access to the decoy datasets generated by the top ten groups in the recent CASP competitions, we can only evaluate the model submitted by a group. Our analysis employs RMSD and GDT_TS. Since the analysis above related HEA-AD superior over other HEA variants, we focus on HEA-AD. Fig. 3 compares the best RMSD achieved by HEA-AD on a CASP target to the RMSD of the model submitted by the top ten performing groups. Fig. 3 shows that HEA-AD ranks in the top ten on $6/10$ targets. The algorithm ranks 1st on two targets, T0859-D1 and T0957s1-D1, 2nd on T0953s1-D1, and 3rd on T0897-D1. Fig. 4 compares GDT_TS values. HEA-AD ranks in the top ten on $3/10$ targets. The algorithm ranks 1st on two targets, T0859-D1 and T0897-D1.

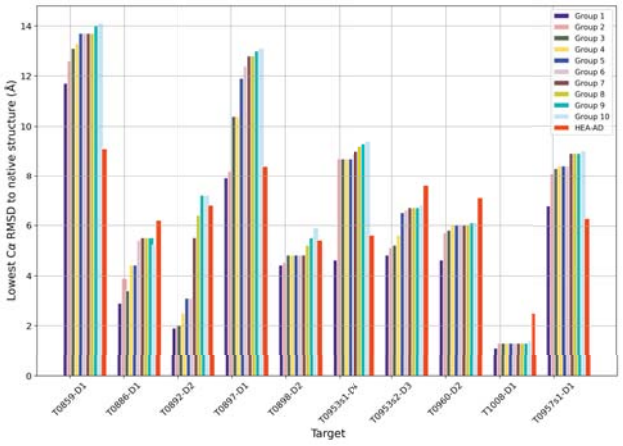

Fig. 3. Comparison of the lowest RMSD (measured in Å) obtained by the top 10 groups and HEA-AD on each target in the CASP dataset.

IV. 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. The results from the RMSD-based analysis agree with these observations. The better balance between exploration and exploitation in HEA-AD yields better-quality decoys in both score and proximity to the native structure. The comparison with top-performing groups over CASP targets shows that HEA-AD is a competitive algorithm on hard CASP targets. These results warrant further

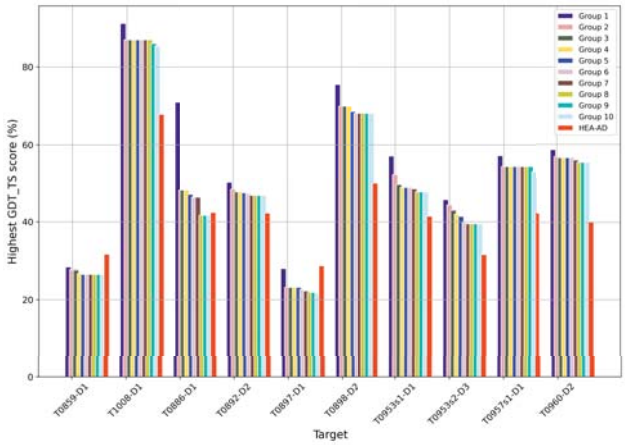

Fig. 4. Comparison of the highest GDT_TS score (measured in percentage) obtained by the top 10 groups and HEA-AD on each target in the CASP dataset.

research on more powerful stochastic optimization algorithms. Future work will investigate the impact of adaptive selection in multi-objective optimization, as well as its interaction with different variation schemes.

ACKNOWLEDGMENT

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 for useful discussions 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] N. Perdigao, J. Heinrich, C. Stolte, K. S. Sabir, M. J. Buckley, B. Tabor, B. Signal, B. S. Gloss, C. J. Hammang, B. Rost, A. Schafferhans, and S. I. O'Donoghue, "Unexpected features of the dark proteome," *Proc Natl Acad Sci USA*, vol. 112, no. 52, pp. 15 898–1590, 2015.

[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.

[4] 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.

[5] C. Zhang, S. M. Mortuza, B. He, Y. Wang, and Y. Zhang, "Template-based and free modeling of I-TASSER and QUARK pipelines using predicted contact maps in CASP12," *Proteins: Struct, Funct, and Bioinf*, vol. 86, no. S1, pp. 136–151, 2018.

[6] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis, "Protein structure prediction using multiple deep neural networks in the 13th critical assessment of protein structure prediction (casp13)," *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 12, pp. 1141–1148, 2019.

[7] A. Shehu, "Probabilistic search and optimization for protein energy landscapes," in *Handbook of Computational Molecular Biology*, S. Aluru and A. Singh, Eds. Chapman & Hall/CRC Computer & Information Science Series, 2013.

[8] 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.

[9] 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.

[10] A. Zaman and A. Shehu, "Balancing multiple objectives in conformation sampling to control decoy diversity in template-free protein structure prediction," *BMC Bioinformatics*, vol. 20, no. 1, p. 211, 2019.

[11] 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.

[12] J. Lee, P. Freddolino, and Y. Zhang, "Ab initio protein structure prediction," in *From Protein Structure to Function with Bioinformatics*, 2nd ed., D. J. Rigden, Ed. Springer London, 2017, ch. 1, pp. 3–35.

[13] J. Hou, T. Wu, R. Cao, and J. Cheng, "Protein tertiary structure modeling driven by deep learning and contact distance prediction in casp13," *Proteins*, vol. 87, pp. 1165–1178, 2019.

[14] A. Zaman, P. Parthasarathy, and A. Shehu, "Using sequence-predicted contacts to guide template-free protein structure prediction," ser. BCB '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 154–160.

[15] B. Olson and A. Shehu, "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.

[16] 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.

[17] 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.

[18] D. Xu and Y. Zhang, "Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field," *Proteins: Struct, Funct, Bioinf*, vol. 80, no. 7, pp. 1715–1735, 2012.

[19] V. Cutello, G. Narzisi, and G. Nicosia, "A multi-objective evolutionary approach to the protein structure prediction problem," *J Roy Soc Interface*, vol. 3, no. 6, p. 0083, 2005.

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

[21] 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.

[22] 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.

[23] A. Zaman and A. Shehu, "Equipping decoy generation algorithms for template-free protein structure prediction with maps of the protein conformation space," in *11th Intl Conf on Bioinf and Comput Biol (BICoB)*, ser. EPiC Series in Computing, vol. 60. EasyChair, 2019, pp. 161–169.

[24] ——, "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.

[25] A. Zaman, P. Kamranfar, C. Domeniconi, and A. Shehu, "Decoy ensemble reduction in template-free protein structure prediction," in *ACM Conf on Bioinf and Comput Biol Workshops (BCBW): Comput Struct Biol Workshop (CSBW)*, Niagara Falls, NY, 2019, pp. 562–567.

[26] 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.

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

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

[29] 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.

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