Beyond the Fold:

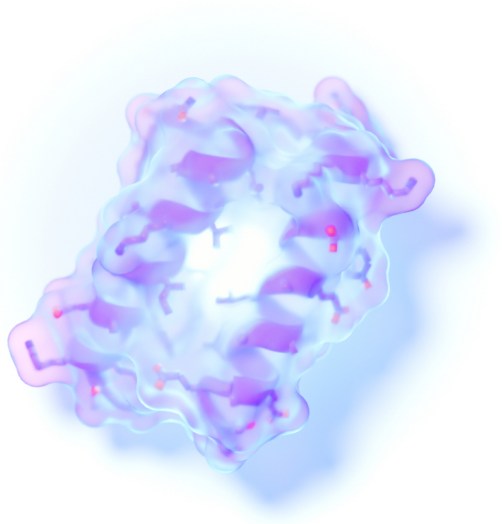
Exploring Uniform Random Sampling in HP Model Protein Folding

submitted in partial fulfillment for the degree of master of science

Jesse Kommandeur
12716197

Master Information Studies
data science
Faculty of Science
University of Amsterdam

Submitted on 21-01-2024



	Supervisor	2nd Examiner
Title, Name	Daan van den Berg	Joost Berkhout
Affiliation	VU Amsterdam	VU Amsterdam
Email	d.van.den.berg@vu.nl	joost.berkhout@vu.nl

ABSTRACT

Protein folding is a fundamental process in biology, with profound implications for understanding biological functions and diseases. The identification of native and stable protein conformations is an integral aspect of this process but is time-consuming to do experimentally, leading researchers towards computational and algorithmic methodologies such as the HP model. In light of the complexities associated with accurately simulating the process of folding proteins, the focus has therefore often shifted towards (meta)heuristics. However, these methods face limitations due to the prevalence of invalid conformations in the search space, posing challenges in achieving uniformly random and unbiased initial populations for algorithms. This paper addresses these issues and raises a twofolded question. The first involves the potential of ensuring uniform randomness in initial populations of metaheuristic algorithms to not bias protein folding simulation and optimization. The second aspect explored the impact of five sampling techniques on the diversity and quality of these initial populations. Experimental results demonstrate a clear disparity in the effectiveness of different sampling methods for generating initial protein conformations. The two break sampling methods require extensive resampling for longer proteins, whereas both backtracking methods, though less dependent on resampling, face computational challenges with increased protein lengths. This highlights a fundamental trade-off between the efficiency of these methods and their adaptability to larger protein sequences. Additionally, this paper identifies a crucial gap in the field: the absence of effective benchmarking methods for assessing the uniform randomness of longer protein sequences. As we progress in sampling proteins of realistic lengths, the exponential growth in possible conformations poses a formidable challenge in assessing these models and approaches. As we observe these challenges of finding uniformly random and unbiased initial populations of conformations, the empirical evidence could suggest that protein folding might be harder than other problems within their NP-hard class.

KEYWORDS

Protein Folding, Uniform Random Sampling, Break Sampling, Backtracking, Solution Space Sampling

GITHUB REPOSITORY

https://github.com/jessekommandeur/Protein-Folding

1 INTRODUCTION

Proteins, complex structures composed of amino acid chains, are crucial in executing a variety of vital processes within the body. These molecules are typically stored in a 'folded' state within cells, a configuration essential for their functional efficacy [21]. However, the misfolding of proteins is implicated in numerous health issues, including cancer, Alzheimer's disease, and cystic fibrosis, and understanding their multi-dimensional structures is essential for deciphering their functions [11, 21, 27].

The artificial synthesis of proteins, a practice initiated in the 1960s, has been pivotal for developing treatments targeting diseases associated with protein misfolding [25]. The identification of native and stable conformations is an integral aspect of this process but is notably time-consuming to do experimentally, leading researchers towards computational and algorithmic methodologies [9].

Yet, the complexity of protein folding, especially in predicting the multifaceted structures that proteins adopt, presents formidable challenges when finding stable(st) conformations due to the number of possibilities into which a protein can fold [6, 10]. Proteins are diverse, with small peptides composed of around 50 amino acids to gigantic structures such as titin, containing more than 27.000 amino acids. The 'common' protein, however, typically consists of approximately 200 to 300 amino acids, with various proteins containing more than 2.000 [5, 31].

This vast array of potential amino acid chains and combinations leads to an exponential number of possible conformations, a scenario that already tests the limits of contemporary computational capabilities [29]. While tools like AlphaFold have made remarkable strides, they are not exempt from delivering off-target folding patterns that diverge from natural occurrences [17, 26].

In the realm of exact algorithms, simplified models like Dill's hydrophobic-polar (HP) folding model have gained prominence [20]. In the HP model hydrophobic amino acids (H) are predisposed to be adjacent, and polar amino acids (P) do not have that preference. When two hydrophobic amino acids lie next to each other, an 'H-bond' is formed due to the attractive forces between the two. And the more bonds, the more stable the protein [20]. Figure 1 demonstrates this concept based on the first chain of the insulin protein, with Figure 1c specifically illustrating the HP model lattice structure with 3 H-bonds between hydrophobic amino acids.

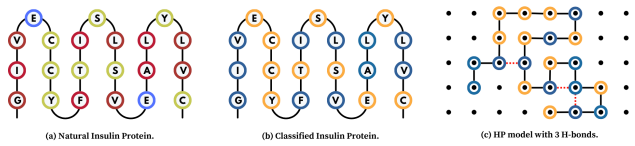


Figure 1: From Protein to HP-model
Transforming a natural protein (insulin) into the HP model.

Protein folding is fundamentally an energy-minimization endeavour. Proteins adopt energetically favourable conformations. However, the NP-completeness and NP-hardness of the protein folding problem, even within the simplified HP model, underscores the computational intensity of identifying or even verifying maximally stable conformations, particularly given the expansive length of proteins [2, 15].

In light of these complexities, the focus has often shifted towards (meta)heuristic methods. These approaches, however, are not without their hurdles [12]. The initiation of metaheuristic algorithms with randomly chosen individuals, a practice advocated by various metaheuristic methods such as Genetic Algorithms, Simulated Annealing, Particle Swarm- and Ant Colony Optimization, could encounter the obstacle of a search space saturated with invalid conformations [16]. The aspiration for uniformly random and unbiased initial populations for algorithms is challenged by the saturation of

the conformation space with invalid structures, raising questions about the adaptability of Genetic Algorithms to the HP-protein folding conundrum [16]. This leads us to the following research questions: *Can uniform randomness in initial populations of metaheuristic algorithms be ensured to enhance accuracy and efficiency of protein folding?* and *What is the impact of different (re)sampling techniques on quality and diversity of initial populations in the context of the HP-protein folding problem?"*

2 RELATED WORK

The HP-protein folding problem, a well-known challenge in computational biology, has seen various methodological approaches since Unger and Moult's seminal work in 1993 as described in [16]. Their study, one of the first to employ Genetic Algorithms (GA) in this domain, compared Monte Carlo methods with GAs, although it left certain technical aspects, like the initialization of uniform random starting populations, undescribed [28]. Subsequent works echoed these limitations, offering valuable but partial insights into the management of invalid conformations and the need for resampling during mutation, crossover, and initialization as we will discuss below.

Patton et al. made significant strides, pioneering a GA that accommodated and penalised collision-rich conformations, thus facilitating quicker identification of lower-energy conformations [24]. The later work of Custodio et al. built upon this as they also enhanced a GA for protein folding by introducing a revised HP model scoring system that favoured natural-like compact structures, and a diverse genetic population through a selection scheme and multiple-point crossover [7]. These modifications improved algorithm performance, but still necessitated extensive evaluations.

On a different trajectory, Bui and Sundarraj concentrated on evolving the secondary structures of the hydrophobic subsection independently, leaning on a robust library of structures [4]. The necessity for numerous repair mechanisms hinted at the nuanced challenge of managing collisions. Lin and Hsieh presented a hybrid model, blending the Taguchi method and particle swarm optimization with GAs, showing improved performance, yet leaving collision treatment and the uniform random sampling of the initial population opaque [22].

A more recent study navigated towards multi-objective optimization, striving for a balance between collision numbers and conformation stability [13]. The research showed that introducing a bias in the optimization criteria towards reducing collisions results in better performance compared to single-objective optimization methods. However, Garza et al. also concluded that their approach resulted in a large state space that is dominated by neutral networks that contained invalid conformations. In a similar vein, a paper by Wang et al. ventured into cloning and 'chaotic mutation' using GAs, yet the collision treatment and initialization of initial populations remained undescribed [30]. Other recent contributions, integrated concepts like hill climbing and quantum GAs but left gaps in presenting substantial data on collision frequency and how random initial populations were sampled [1, 3].

The review consistently unveils a recurring constraint: the lack of focus on sampling initial populations randomly for protein up to length 100, which is less than half of the average protein length [16, 31]. This aspect underscores the inherent challenge of random sampling and raises questions on the broad applicability of iterative algorithms for extensive HP-protein folding scenarios. In addition, a recent paper by Jansen et al. spotlights two contrasting strategies - strict inclusion of valid solutions and a more lenient approach permitting but penalizing collisions [16]. This dives into a possible fundamental issue of uniform random sampling folding and emphasizes finding deterministic time algorithms for constructing initial conformations, ideally without iterative resampling techniques.

Their experiment, involving the random folding of 'neutralized' proteins, reveals a linear increase in collision numbers with protein length, casting doubts over the feasibility of uniformly sampling valid initial solutions for longer proteins [16]. This insight propels the discourse beyond the technical and methodological terrain navigated by previous researchers, to the foundational quagmire of sampling initial populations for protein folding. It accentuates the trade-off between computational capability and the ability to simulate proteins and their folding process.

Furthermore, the work of Jansen et al. questions whether conformations generated using the HP model in both 2- and 3D approximate empirical evidence of a uniform random sample distribution [16]. This means that if all valid conformations were to be enumerated, the probability of selecting any one of these conformations would have the same chance of getting chosen. This is of significant importance since GAs, hill climbing, or simulated annealing inherently have the potential to initiate from a uniform random starting condition [12]. This observation is intriguing, particularly because the literature suggests a lack of attention or consideration given to this aspect by those employing GAs within the realm of protein folding. Yet, this is a critical step; if the selection of an initial conformation is not from a uniformly random sampling distribution, there exists a probability of consistently exploring only a subset of the solution space and therefore locating the same local optima.

Additionally, for certain problems such as the travelling salesman problem (TSP), it is easier to employ uniform random sampling. One can select a specific number of random cities, sequence them randomly, and be assured of a uniformly random solution. Therefore, random solutions of the TSP can be easily sampled by placing n cities on an area A uniformly at random [14]. However, this appears to be an unattainable approach in protein folding due to order and spatial constraints, possibly making protein folding harder than TSP.

Consequently, an approach undertaken in a previous study involved evaluating the frequency of required resampling to attain a valid conformation without violations [16]. The advantage of this method is its adherence to uniform randomness. However, a significant drawback is the extensive repetition of resampling, particularly since the likelihood of encountering a violation, especially for large proteins, is exceedingly high. While resampling can potentially be performed with backtracking, it remains uncertain if this yields empirical evidence of uniform random sample distribution.

However, in the context of optimization and search methods, it is worth mentioning that GAs, hill climbing, and simulated annealing do not necessarily have to begin from a uniform random starting condition, but research has shown that it is beneficial for them to do so [8, 23]. Starting with random initial conformations can help these algorithms explore the solution space more thoroughly

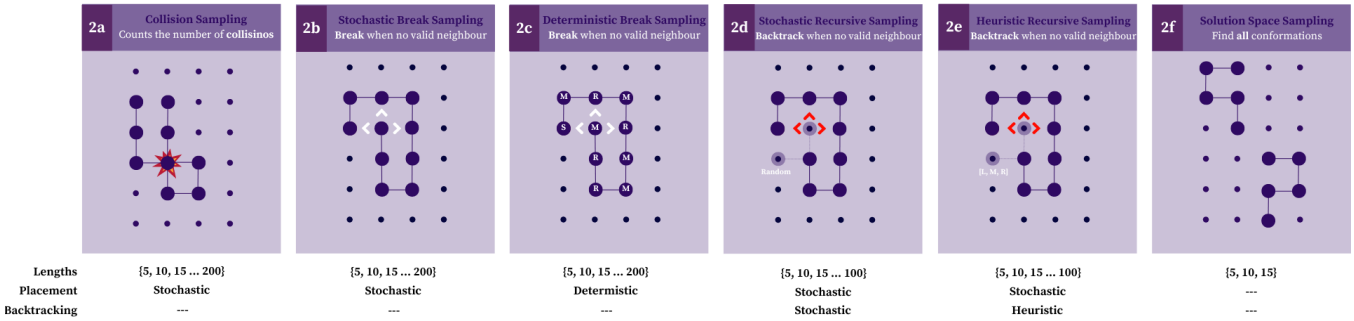


Figure 2: Sampling Methods

Figure 2a refers to the collision sampling as demonstrated in [16]. Figures 2b and 2c refer to the stochastic and deterministic break sampling methods. Figures 2d and 2e illustrate stochastic and deterministic recursive sampling, while 2f shows two conformations of the solution space for a protein of length 5.

and avoid premature convergence on local optima thus finding a more optimal or global solution. Furthermore, a lack of uniform randomness could potentially bias the exploration and trapping of the optimization process into local optima.

The mix of insights from these studies paints a rich, multifaceted landscape of HP-protein folding. The journey from the pioneering works of Unger and Moult to the more recent insights of Jansen et al. unveils a narrative filled with complexities [16, 28]. The nuances of GA application, collision management, and the foundational enigma of sampling initial populations pool into a compelling intersection that sets the stage for future research.

In this paper, we will start by extending the work of Jansen et al. as shown in Figure 2a [16]. Instead of focusing on including valid solutions and a lenient approach permitting but penalizing collisions, we experiment with both deterministic and stochastic sampling methods that will break once we encounter collisions for different protein lengths, as shown in Figures 2b and 2c. Subsequently, we deploy two backtracking algorithms, as shown in 2d and 2e, to discover to what extent this could produce empirical evidence of a uniform random distribution for the initial populations of proteins before we start folding. In our final experiment, we use a stacked depth-first-search (DFS) approach to sample the solution space, which is later used as a benchmark to assess the sampling results of the other four experiments.

3 METHODOLOGY

3.1 E1 - Stochastic Break Sampling

In our first experiment, we performed random folding of 1,000 'neutralized' 1D strings of proteins, for lengths {5,10,15,...,95,200}, on a 2D lattice, void of any prior assumptions. This 2D lattice was designed to be twice the size of the maximum amino acid sequence to ensure enough space for the protein to fold. We termed the process 'neutralization' because it disregards the hydrophobic and polar labels from the amino acids as shown in figure 1. This underscores our focus on valid conformations over optimal ones, aligning with the methods described in previous work [16].

Our methodology began with the central placement of the first amino acid on the grid which is denoted with *Start (S)*, since it has no relative direction towards other acids. From here, the sequential placement of amino acids commenced. Each amino acid was positioned one after another, adhering to a chain-relative representation. Specifically, each subsequent placement randomly selected a position from the set {*left (L), middle (M), right (R)*}, while diligently avoiding any backward or neighbouring collisions by checking empty adjacent grid locations. This chain-relative approach ensured that every subsequent amino acid was next to the preceding one. We maintained a count of the total number of amino acids (n_{aminos}) each time an amino acid was positioned on the 2D lattice. This process was iterated upon until no valid adjacent locations remained, as visualized in Figure 2b.

3.2 E2 - Deterministic Break Sampling

In experiment 2, we conducted deterministic folding of 1,000 predefined strings of proteins with lengths {5,10,15,...,95,200}, on a 2D lattice sized to twice the length of the maximum protein sequence. However, this second approach uses predetermined folding patterns, rather than random placement at each step, for each amino acid in the sequence. The predetermined directions were set to either {L, M, R} for each amino acid, ensuring the path of the chain was decided before the simulation began. For example, the protein in Figure 2c has a predefined folding pattern of the following path: {S, M, R, M, R, M, M, R, R, M}. This method eliminates randomness in the placement and follows a predetermined sequence that is unique for each protein string but consistent across trials.

The protein chain was extended by positioning each amino acid adjacent to the previous one according to its predetermined direction. This approach also avoids backward movements like experiment 1, but is more sensitive to collisions, since its path is predetermined and thus less flexible in avoiding adjacent neighbours, which might result in a collision.

This deterministic chain-relative approach guarantees that the predetermined folding pattern is achieved for every simulation unless a collision is encountered along its path. The total count of amino acids was tracked as the chain unfolded on the grid. The folding process continued until all amino acids were placed on the grid or the predetermined adjacent placement location resulted in a collision, exemplified in Figure 2c.

3.3 E3 - Stochastic Recursive Sampling

In our third experiment, we mirrored the initial steps of experiment 1. A similar set of 1,000 'neutralized' 1D strings of proteins were randomly folded on a 2D lattice, adhering to the same conditions but with protein lengths {5,10,15,...,95,100}, due to the computational- and time complexity of recursion. Unlike experiments 1 and 2, a significant divergence was introduced when no valid adjacent locations remained. Rather than concluding the process, we implemented backtracking to navigate through previously explored paths and explore alternative configurations.

The backtracking approach was algorithmically structured to maintain the inherent sequentiality of amino acid placements. Starting with the central positioning of the first amino acid, the sequence continued to unravel by evaluating potential neighbours for subsequent placements. However, it's crucial to note that this wasn't a linear journey. When an amino acid's placement resulted in an impasse, the algorithm retraced its steps to the last valid position, thus allowing us to probe different, previously uncharted, paths as shown in Figure 2d.

This recursive nature, where the algorithm continually revisited amino acid placements, was intrinsic in its ability to expand the scope of our exploration. By randomizing the order of the neighbours at run-time, we ensured that every run could potentially lead to different conformations, even if the starting points were the same. Note that these random backtracking steps were not predetermined before the placement of amino acids on the grid, so every acid has a randomized backtracking order from the combination {L, M, R}, which has 6 (3!) permutations.

3.4 E4 - Heuristic Recursive Sampling

Experiment 4 used the initial conditions of experiment 3, utilizing a set of 1,000 'neutralized' proteins for lengths {5,10,15,...,95,100}, folded on a 2D lattice. In this experiment, however, we diverged from stochastic methods by introducing a more deterministic backtracking method. Upon reaching an impasse where no further placements were valid, the deterministic approach used a heuristic backtracking rule that can be expressed using propositional logic: $L \vee (\neg L \wedge M) \vee (\neg L \wedge \neg M \wedge R)$. This rule states that left L is the first preference. If L fails ($\neg L$), then try middle M as the next option. If both L and M fail ($\neg L$ and $\neg M$), then try right R.

The backtracking continued in this structured manner, systematically exploring the next available direction in the predetermined sequence whenever an impasse was encountered. The recursive algorithm ensured adherence to the sequence of directions predetermined for each amino acid, retracing steps in a heuristic manner whenever the folding path reached a dead end.

3.5 E5 - Solution Space Sampling

In experiment 5, we generated the complete solution space to enumerate all valid conformations of amino acid sequences with lengths {5, 10, 15}. This comprehensive approach served to benchmark the empirical uniform randomness of the other four sampling methods utilized in the preceding experiments. Using a 2D lattice scaled to accommodate the longest protein sequence, we systematically generated each possible conformation for the given lengths. A combinatorial approach was employed to conduct a stacked DFS algorithm across the conformational landscape, assuring that each unique conformation was accounted for once and only once.

The protein chain unfolding commenced from the grid's central point, with each amino acid conforming to a predefined direction sequence {L, M, R}. Therefore, mathematically, the conformation space C can be defined as shown in (1), where n represents the number of amino acids, and each d_i denotes the direction of the i-th amino acid, except for the first amino acid which is always 'S'.

$$C = \{(S, d_2, d_3, \ldots, d_n) \mid d_i \in \{L, M, R\} \text{ for } i = 2, 3, \ldots, n\} \quad (1)$$

To cover all possible sequences of directions for the amino acids in a two-dimensional space, we could use formula (2), where n represents the number of amino acids. Note that the number of conformations in a three-dimensional space could be calculated by replacing 3 with 5.

$$C_{total} = (1 \times 1 \times 3)^{n-2} \quad (2)$$

Following the initial definition of the conformation space C and C_{total}, it is important to recognize that not all sequences within this space represent viable protein structures. These conformations include configurations where amino acids overlap or contain collisions [16]. Therefore, to accurately represent the valid solution space C_{valid} for a given amino acid length, one must exclude these invalid sequences as shown in formula (3), where invalid conformations contain all unfeasible directional sequences from C_{total}.

$$C_{valid} = C_{total} - \text{invalid conformations} \quad (3)$$

Upon refining the conformation space C_{total} to the valid solution space C_{valid} by excluding unfeasible sequences, we encounter another aspect to consider: mirrored conformations. These are different directional sequences that lead to identical two-dimensional structures, which are visually shown in Figure 2f. These two conformations have different paths, but are mirrored equivalents. To accurately reflect the unique protein structures, it's essential to eliminate these mirrored duplicates. This is accomplished by dividing the total number of valid conformations in half, acknowledging that each structure typically has a 'mirrored duplicate'. The formula for the number of unique samples C_{unique} is therefore given as (4).

$$C_{unique} = \left(\frac{C_{valid} - 1}{2}\right) + 1 \quad (4)$$

The dataset generated from the stacked DFS algorithm provided the basis for benchmarking the sampling efficacy of the previous experiments. By comparing the conformational distribution obtained from the combinatorial search with those from the sampling methods, we aimed to evaluate their performance in producing an empirical uniform random sample distribution. This analysis is crucial for understanding the randomness of our sampling methods relative to the entire landscape of possible conformations.

The number of conformations C_{total}, valid conformations C_{valid}, and unique conformations C_{unique} for lengths {5, 10, 15} are given in Table 1. Note how calculating the solution space scales exponentially or worse due to the combinatorial explosion of the DFS algorithm on the solution landscape.

Table 1: Solution Space Sampling Descriptives

n	C_{total}	C_{valid}	C_{unique}
5	27	25	13
10	6561	4067	2034
15	1,594,323	593,611	296,806

4 EXPERIMENT

4.1 Data Generation

Following the steps for each experiment as described in our method section, we dedicated a focused approach to generating datasets that would effectively capture the outcomes of initial protein conformations. For each experiment, we generate 1,000 samples for varying protein lengths. The maximum length was based on the time complexity of each experiment since both break sampling methods run in polynomial time, while backtracking and solution space sampling both have a time complexity that runs in non-polynomial time. An overview of the datasets that were generated is described in Appendix B

Experiments 1 to 4 were each run five times, each run tagged with either {a, b, c, d, e}. Each line in Figure 3 traces the average time it takes to produce 1,000 samples for each algorithm across different lengths of amino acid chains. The varying trajectories reflect the time complexity associated with each experiment.

Notably, while the increase in processing time increases linearly with the amino acid length for experiments 1 and 2, the time to generate samples increases exponentially for longer protein samples in experiments 3 and 4 due to the higher number of recursions as we will discuss in the results section. The increasing difference as the protein length elongates both emphasizes and validates the differences in time complexity between the break sampling- and backtracking methods. A more detailed outline of the individual timing of each experiment is described in appendix C.

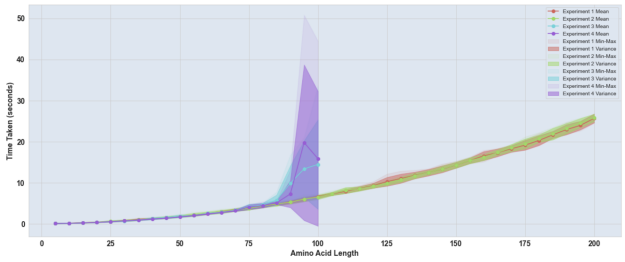


Figure 3: Experiment timing
Relationship between sample size and sampling time across four experiments. All experiments shows an increase in time as proteins get longer, with experiments 1 and 2 reflecting a polynomial-, and 3 and 4 a non-polynomial growth for longer proteins.

4.2 Experiment Validation

After generating all datasets, we verified whether our simulations were consistent among trials and not due to some form of randomness. Therefore, in experiments 1 and 2, our goal was to understand whether the average number of amino acids successfully placed on the grid remained consistent across multiple runs. To this end, we performed an ANOVA, comparing the sample means of five runs with varying amino acid lengths. This test helped determine if there were significant differences in the sample means by comparing the variance across trials, indicating inconsistency in the sampling process.

Given the datasets for experiment 1 and experiment 2, each representing a separate run of the experiments, we defined H_0 as there being no significant difference between the means of different experimental runs. H_1 posited that at least one out of five trials had significantly different means. By comparing the calculated test statistic to the critical value at a 95% confidence level and observing p-values below the 0.05 alpha threshold, we identified any significant discrepancies in sample means between trials.

In experiments 3 and 4, we employed a similar ANOVA test to analyze the stability of the recursion count across trials. This statistical measurement compared the number of recursions for each of the five datasets within both experiments. H_0 presumed no difference in recursion proportions across datasets, while H_1 suggested variation. A test score and corresponding p-value for each dataset comparison informed us of any significant differences, with p-values lower than 0.05 indicating a rejection of the null hypothesis. A summary of all validation test results is shown in table 2. A detailed outline of the experimental validation is given in Appendix D.

Table 2: Experiment Validation Statistics

$Experiment$	$F-value$	$p-value$	H_0	H_1
1	0.703	0.606	10	0
2	1.122	0.414	9	1
3	0.907	0.526	9	1
4	0.930	0.485	10	0

4.3 Experiment Features

Beyond the various placement methods of amino acids on a 2D grid, all experiments were designed to gather descriptive data about all protein samples to keep track of their sampling behaviour. In total, we logged 14 features across all experiments, encompassing aspects like the *shape* and $H-bonds$ of each protein. These features form the bakermat for our result section, and a short description of each feature is listed below. The datasets, spruced with their features, can be further explored in Appendix A, Github or Kaggle [18, 19].

(1) **Amino Acid Length**: Number of amino acid residues in a given protein sequence.
(2) **Num Hydrophobic**: Number of hydrophobic amino acids in the protein sequence.
(3) **Num Polar**: Number of polar amino acids in the protein sequence.
(4) **1D protein**: Primary structure of the protein, which is the linear sequence of amino acids.
(5) **2D protein**: Secondary structure of the protein, folded on a 2D Numpy grid.
(6) **Amino Acids on Grid**: Number of amino acids on the 2D grid.
(7) **Trimmed 2D protein**: Trimmed representation of the 2D grid removing redundant information from the grid.
(8) **Shape 2D protein**: 2D shape of a protein on the grid, giving insights into the protein's topology in two dimensions.
(9) **Amino Acid Order**: Order position in which amino acids are placed on the grid.
(10) **Amino Acid Direction**: Direction of each amino acid placement relative to the previous one on the grid.
(11) **H-Bonds**: Number of hydrogen bonds within the protein structure. Hydrogen bonds refer to the stability of both protein structures.
(12) **H-Ratio**: Ratio of hydrophobic bonds to the number of amino acids on the grid.
(13) **Recursions**: Number of recursions needed for a valid protein conformation, only used in our recursive experiments.
(14) **Time Taken (s)**: Number of milliseconds it takes to produce protein sample.

5 RESULTS

5.1 Experiment 1 and 2

In the first two experiments, both algorithms randomly placed 1,000 amino acids on a 2D grid, for lengths {5, 10, 15, ..., 200} without backtracking. The range for n_{aminos}, representing the number of amino acids on the grid in any given conformation, was established with a minimum value of $n_{\text{aminos}} = 5$ for experiment 1 and $n_{\text{aminos}} = 4$ for experiment 2 due to circular collision occurring as a result of a predetermined placement path. The maximum value equates to the number of amino acids on the grid that can be placed without encountering a collision. In Figure 4, for each specified amino acid length, we documented the mean quantity of n_{aminos} that were successfully placed on the grid for each of the 1,000 conformations generated. This process was conducted for both experiments 1 and 2. A notable trend observed in both experiments is the decrement in the number of valid conformations concomitant with an increase in n_{aminos}. This phenomenon is attributed to the escalating likelihood of collisions as the protein chains elongate, reducing the valid conformational space.

For experiment 1, the frequency of valid conformations remains high for n_{aminos} less than 20. As n_{aminos} increases to 100 and 200, the percentage of valid conformations drops markedly, to 22.7% and 3.4% respectively. This reduction suggests a decrease in available conformations is due to collisions. The probability of sampling a zero-violation conformation in experiment 1 is shown in formula (5).

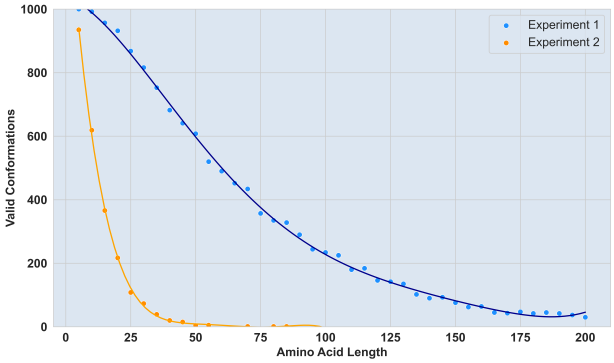


Figure 4: Valid conformations per amino acid length
Distribution of the valid conformation per amino acid length. For experiment 1, the frequency of valid conformations remains high for n_{aminos} less than 20. Experiment 2, in contrast, exhibits an exponential decrease in the number of valid conformations as proteins increase.

$$P_{(\text{violations}=0)} = 1227.38 \cdot e^{-0.02 \cdot n} \tag{5}$$

Experiment 2, in contrast, exhibits an exponential decrease in the number of valid conformations as proteins increase. This decline implies that to achieve a comparable number of valid conformations to experiment 1, significantly more resampling would be necessary in experiment 2. The steeper drop-off observed in experiment 2 reflects the increased probability of collisions as the protein chains lengthen, a pattern similar to the experiment of [16]. The probability of sampling a zero-violation conformation in experiment 2 is shown in formula (6).

$$P_{(\text{violations}=0)} = 1556.88 \cdot e^{-0.10 \cdot n} \tag{6}$$

The differing resampling requirements between the experiments can be attributed to the algorithms' approach to collisions. In experiment 1, when the stochastic selection of the next neighbour results in a collision, the algorithm actively searches for neighbouring configurations that do not lead to a collision. Thus, a collision is only confirmed if no collision-free neighbours can be found, which inherently reduces the frequency of invalid conformations. This approach contrasts with experiment 2, where the initial choice is final, and if it leads to a collision, the conformation is immediately deemed invalid without further neighbour checks.

Figure 5 shows the distribution of valid protein conformations in experiment 1, binned within distinct length intervals: 5-50, 55-100, 105-150, 155-200. Firstly, there is a noticeable variability in valid conformations as protein length changes. With increasing protein length, the distribution widens, highlighting a rise in structural variability in elongated amino acid chains. This is evident in the shift of the peak frequencies across the lengths. While proteins of length 50 display a pronounced peak close to their maximum extent, indicating a majority of conformations nearing the full length, the peak's position migrates leftward for longer proteins. This suggests a diminishing proportion of conformations that utilize their complete length without encountering a collision, especially as the sequence lengthens.

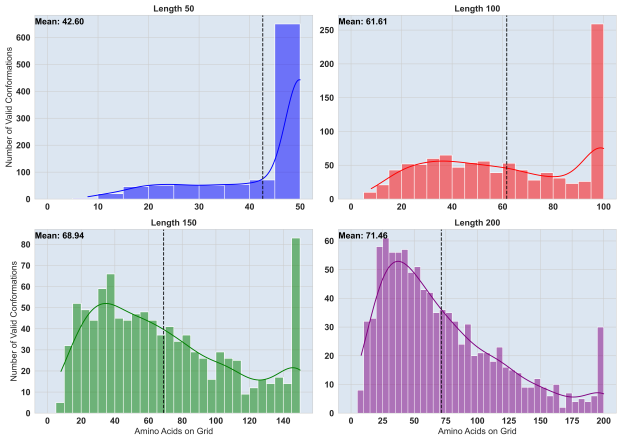


Figure 5: Distribution of valid conformations
Number of valid protein conformations across different lengths. Shorter lengths exhibit peak formations near their respective maximums, while longer proteins show a broader distribution, underscoring the structural complexities with increased length.

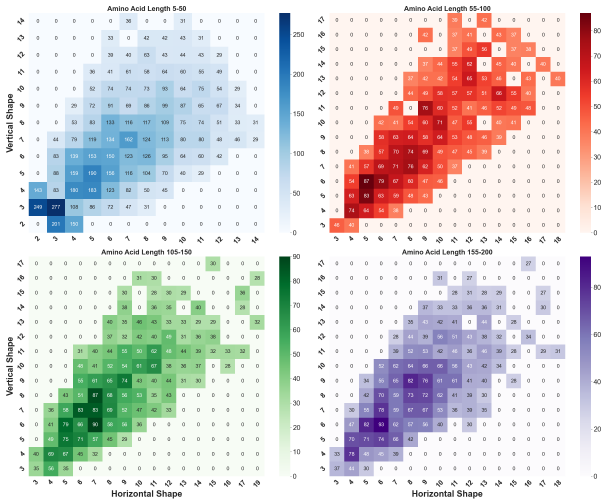


Figure 6: Distribution of shape combinations
Heatmaps illustrating shape combinations for different protein lengths. Shorter proteins manifest concentrated shape patterns, while longer ones exhibit dispersed combinations, highlighting diverse structural possibilities as proteins elongate.

The embedded mean values in each histogram segment of Figure 5 present another layer of interpretation. These averages hint at the typical compactness inherent to protein structures for the indicated lengths. Remarkably, the distributions associated with longer proteins, particularly those with lengths of 150 and 200, appear more dispersed.

Transitioning to Figure 6, we mapped the quantity of valid conformation and their shape combinations for the same amino acid length intervals: 5-50, 55-100, 105-150, 155-200. A shape combination is a combination of the horizontal- and vertical shape of a protein, The closer these numbers are together, the more compact the initial protein conformation (e.g. compact [5,5] vs extended [10,2]). Amongst the 40.000 protein samples were 2.614 unique shapes. For interpretability purposes, we focused on the top 100 dominant shapes, which have the highest frequency and cover 54.05% of all proteins in the dataset.

The representation highlights that shorter proteins manifest higher quantities of certain shape combinations, particularly those in the 5-50 amino acids range. This could suggest that specific shapes could be inherently more dominant or recurrent in shorter proteins when sampling. Furthermore, as the amino acid length advances, there is a slight dispersion in shape combinations.

Another observation from these heatmaps is the variable intensity. The first heatmap's values peak around 250, whereas subsequent maps taper off to about 100 or marginally above. This discrepancy might indicate the reduced frequency of proteins in the dataset as amino acid length augments, or it could indicate that shape combinations become more uniformly distributed across shapes. The former notion harmonizes with the decrement in valid conformations correlated with amino length elevation, as outlined previously.

5.2 Experiment 3 and 4

In our third and fourth experiments, two different algorithms randomly placed 1,000 amino acids on a 2D grid, for lengths {5, 10, 15, ..., 100}. What distinguishes these experiments from the first two is that both stochastic- in experiment 3 and heuristic backtracking in experiment 4 were applied upon encountering a collision. Figure 7 shows the number of recursions for experiments 3 and 4. Although the backtracking approach is slightly different, both experiments show a similar pattern in terms of recursions.

Figure 8 shows the hardness distribution (HD) for the number of recursions for both experiments based on the 25 hardest instances for specific amino acid length intervals. As the HD lines show, the number of recursions needed to generate a valid conformation increases as the number of amino acids grows in both experiments, especially for larger proteins

Alongside the general increase in recursions, the HD also showcases a heightened sensitivity to outliers as protein length extends. The HD distributions for longer amino acid intervals display a more dispersed distribution of recursions for both experiments. This dispersion suggests that longer proteins not only require more recursion on average but also present a significant chance of experiencing an extreme number of recursions. Such outliers could possibly indicate the intricate folding landscape where a few protein conformations deviate substantially from the common folding pathways, requiring an atypically high or low number of recursions to reach a valid folded state.

5.3 Experiment 5

In experiment 5, we aimed to enumerate all valid conformations for protein sequences for protein lengths 5, 10, and 15, leveraging a combinatorial approach to sample the entire solution space.

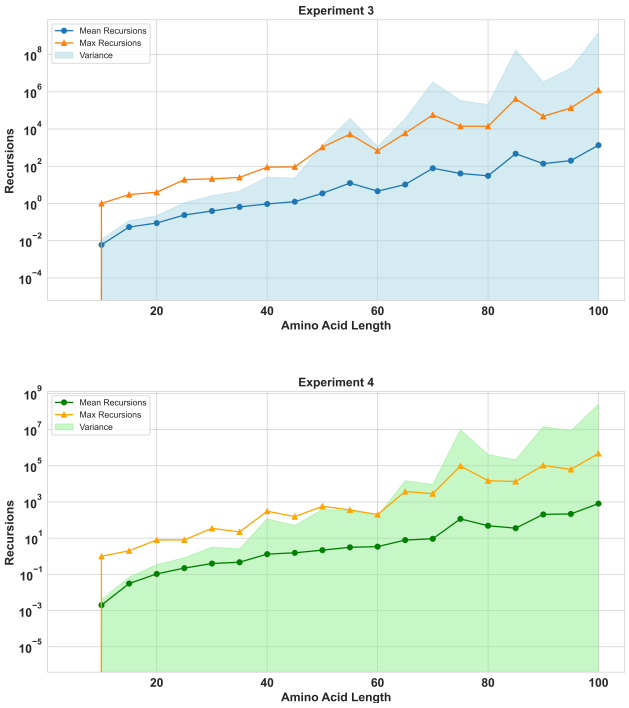


Figure 7: Distribution of recursions

Distribution showing the number of recursions for experiments 3 and 4. Both experiments show similar patterns, since shorter proteins have fewer recursions, while longer ones exhibit extensively more recursions, validating the time complexity as proteins elongate.

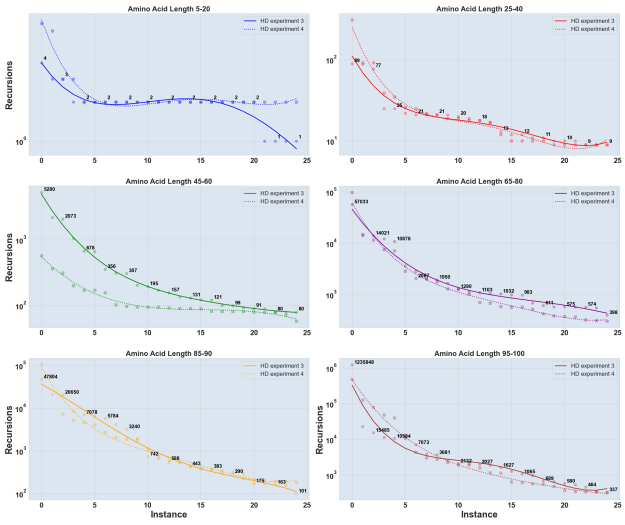


Figure 8: Hardness distribution of recursions

Distribution showing the 25 hardest protein instances in terms of recursions across different intervals. Both experiments show a similar pattern, since shorter proteins have fewer recursions, while longer ones exhibit extensively more recursions, validating the time complexity as proteins elongate.

The distribution of the protein samples for length 5 is described in Figure 9, which matched the frequency of conformations within our four experiments to all possible conformations in the solution space of experiment 5 based on their path. For proteins of this length, we already saw in Table 1 that the solution space is relatively small, with only 25 valid conformations out of a possible 27. This is also illustrated in Figure 9 as there are only two amino acid directions that contain a circle collision: SMLL(L) and SMRR(R). They are mirrored equivalents and only appear in experiment 2, since all other algorithms avoid collisions if valid neighbours are available.

Since the frequencies in Figure 9 could include randomness due to the number of samples, we also isolated experiment 1 to see if we can find empirical evidence for a uniform random frequency distribution. Utilizing this sampling methodology, we generated 1,000, 10,000, 100,000, and 1,000,000 initial conformations for protein length 5. Ideally, with an empirical uniform random sampling distribution, each unique conformation would represent 4% of the total, appearing 40, 400, 4,000, and 40,000 times, since there are only 25 valid conformations.

The distribution of samples for length 5 can be seen in Figure 10, which illustrates the relative frequency of each valid conformational path for the different sample size. Despite increasing sample sizes as shown in the subsequent panels for 10,000, 100,000 and 1,000,000 samples, the trend line does not align with the uniform distribution and their difference is highlighted by the orange area, reflecting the intrinsic difficulty in attaining uniformity within the complex protein folding landscape, already for proteins of length 5.

As we consider proteins of length 10, the situation changes dramatically. The valid conformation count escalates to over 4,000 as shown in Table 1, rendering the solution space sampling method impractical for benchmarking with our current experimental setup. To maintain the same sampling ratio as observed for length 5 (40:1), we would require more than 160,000 samples. While generating this number of samples is within the realm of possibilities for the break sampling methods, it becomes a much more daunting task for the backtracking methods due to their non-polynomial time complexity, especially for longer proteins.

The backtracking process is heavily influenced by the increased length of the protein, leading to an exponential rise in computational time and resources required to sample the solution space adequately. This challenge is not merely a matter of processing speed or efficiency; it is a fundamental limitation imposed by the exponential nature of the problem. As such, the protein solution space sampling ceases to be a viable method for benchmarking the randomness and uniformity of our sampling approaches beyond a protein length of 5 with the current experimental design.

These findings suggest that for protein lengths greater than 5, we should find ways to generate more valid protein conformations faster or alternative benchmarking strategies must be explored. These may include statistical comparisons with theoretical distributions or utilizing advanced computational resources capable of handling the significant requirements of backtracking algorithms for longer protein chains.

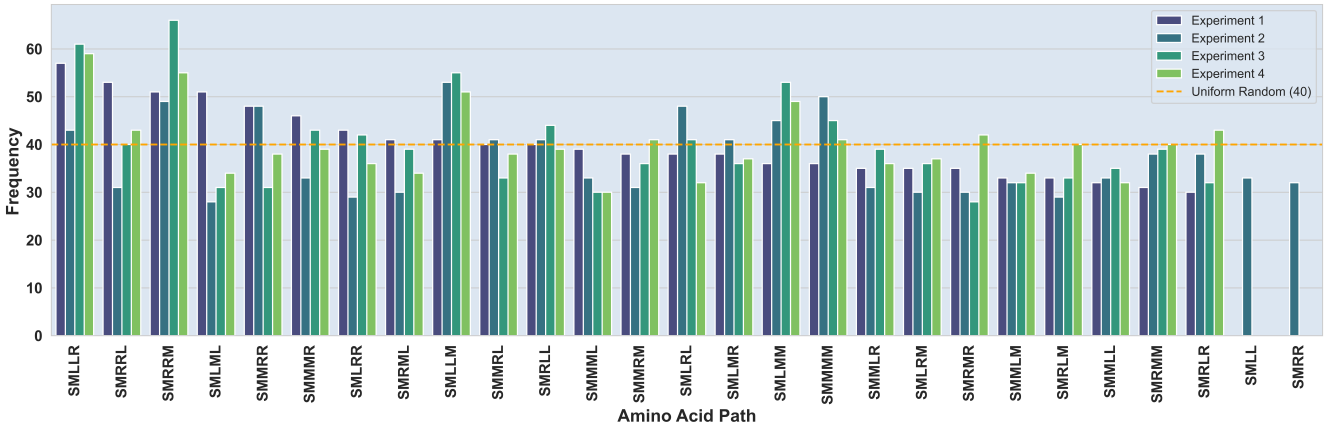


Figure 9: Sample Distribution by Experiment

Bar charts showing the frequency of amino acid paths across the first four experiments, combined with all paths in the solution space generated in experiment 5. A dashed orange line marks the expected count for a uniform distribution (40). Each bar's height indicates the observed frequency, and the trend line highlights deviations from the expected uniformity. The experimental results match the C_{total} (27) and C_{valid} (25) values for n=5 from Table 1.

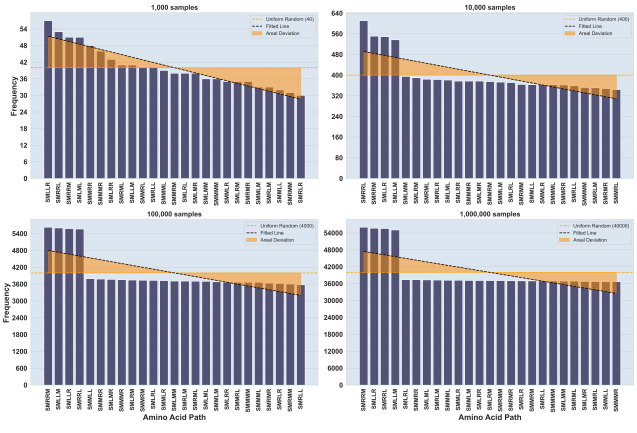


Figure 10: Sample Distribution Experiment 1

Bar chart showing the frequency of amino acid paths for experiment 1. The orange line indicates a uniform distribution and the black trendline fits the data. The orange area reveals the deviation from uniformity.

6 DISCUSSION

6.1 Resampling Versus Recursion

In the context of sampling initial protein conformations, particularly for longer protein sequences, the necessity for extensive resampling poses significant challenges. Our observations in experiments 1 and 2, which employed break sampling methods, underscored this challenge. We noted that these methods required a substantial amount of resampling to achieve valid conformations, particularly as the protein length increased. This possibly indicates that while break sampling methods can be effective, their practicality diminishes with longer protein sequences due to the escalating need for resampling, as also demonstrated in [16]. However, we also showed that the number of valid conformations drops less steeply with the stochastic break sampling approach in experiment 1 compared to the deterministic approach in experiment 2.

One approach to address this challenge is to develop more efficient sampling algorithms that can reduce the number of resamples. Incorporating adaptive strategies that dynamically adjust the resampling rate based on the characteristics of the protein sequence being initialized could be a promising direction. Machine learning techniques, for instance, could potentially be employed to predict the likelihood of invalid conformations, thereby reducing the need for resampling. However, this can also impact the uniformity of our initial populations, as machine learning algorithms can also overfit when exposed to conformations that commonly occur and when not properly tuned.

In contrast, in experiments 3 and 4, which involved stochastic and heuristic recursive sampling, we observed that while there was no need for resampling, the number of recursions increased for longer protein lengths. This increase in recursions, although eliminating the need for resampling, presented its own set of challenges, especially for longer proteins. The computational capacity and time required for these recursive methods also escalated with protein length, indicating a trade-off between the need for resampling and the number of recursions. Additionally, it remains unclear whether their distributions are uniform.

Another avenue to explore could be the development of algorithms that inherently produce fewer invalid conformations. Such algorithms would reduce the dependency on both resampling and extensive recursion, potentially making them more suitable for longer protein sequences. Exploring hybrid approaches that combine elements of different sampling methods might yield more efficient resampling strategies and manage the increased recursion requirements in a more computationally feasible manner. The development of more targeted sampling methods, which can intelligently navigate the solution space to focus on more promising regions, could also be beneficial. However, this targeted approach could on

the other hand also reduce the amount of randomness and therefore the chances of finding a uniform random sampling distribution.

6.2 Benchmarking Limitations

Furthermore, our research identified critical gaps in benchmarking the uniform randomness of our empirical output and in handling the computational limits of solution space exploration, particularly for longer protein sequences. The challenge is twofold: the exponential increase in possible initial conformations with longer sequences complicates the evaluation and comparison of different sampling algorithms, and the exhaustive enumeration of protein folding patterns becomes computationally intensive.

To address these issues, one potential approach is the development of theoretical models that provide a baseline for expected outcomes. These models could help in measuring the uniform randomness of initial populations generated by various sampling algorithms. Additionally, leveraging high-performance computing resources, including distributed methods, could be a solution to the significant computational demands of these algorithms.

6.3 Mirroring Bias

One of the other notable observations from the recursive sampling experiments is the lack of significant differences in the outcome between the two methodologies. This phenomenon can be rationalized by considering the nature of mirrored shape combinations that emerge during the folding process.

In both stochastic and heuristic recursive sampling, the heuristic of choosing between left, middle, and right directions for placing the next amino acid results in mirrored conformational outcomes. Essentially, a left-middle-right heuristic is equivalent to a right-middle-left sequence when backtracking, resulting in a mirrored version of the same conformational shape.

This mirror effect implies that regardless of the backtracking direction, there is always a higher likelihood of making a '90-degree' recursion as opposed to a straight recursion when backtracking stochastically. Statistically, this gives us a 2/3 probability of choosing a 90-degree recursion as the first backtrack option. If this option leads to an invalid placement, there is another 1/2 chance of making a 90-degree recursion over a straight one. This inherent bias towards 90-degree recursions makes them a more dominant feature in the folding landscape. And since we used a left, middle, right, heuristic for backtracking, this could potentially explain why the results from both experiments exhibit remarkable similarities.

While it is challenging to provide empirical proof of this mirroring effect due to the complexity of the protein folding process and the probabilistic nature of the recursions, the observed pattern suggests a logical underpinning and could be worth researching in future work. The prevalence of 90-degree turns in the recursive sampling's backtracking steps could be a key factor in shaping the folding pathways and the resulting conformational space explored by both methods.

7 CONCLUSION

This paper addressed two pivotal questions in the field of protein folding using the HP model. The first question addressed the potential of ensuring uniform randomness in initial populations of metaheuristic algorithms: Can uniform randomness in initial populations of meta-heuristic algorithms be ensured to enhance the accuracy and efficiency of protein folding?

Due to the various challenges in both the break- and recursive sampling methods, make us tend to answer this question with no, the uniform randomness in initial populations cannot be ensured. However, the honest answer to this question is that we don't know, because we cannot check the distribution of samples for realistic protein lengths due to the combinatorial explosion of the solution space. And even if we find methods to get the sample distribution and test whether the empirical samples follow a uniform random distribution, we still face the challenge of generating enough samples of realistic length to fit into this distribution. Our current experimental setup shows that this is not possible due to the number of resamples for both experiments 1 and 2, and the exponential growth in the number of recursions for experiments 3 and 4.

The second question explored the impact of various (re)sampling techniques: What is the impact of different (re)sampling techniques on the quality and diversity of initial populations in the context of the HP-protein folding problem?

Our study demonstrates a notable disparity in the efficacy of these techniques. The findings highlight the necessity of extensive resampling in the two break sampling methods to achieve valid conformations, which becomes increasingly challenging as protein length approaches these real-world instances. While backtracking methods reduce the need for resampling due to their recursive nature, they are not without challenges as well. Specifically, the number of recursions increases exponentially with longer proteins, presenting a significant computational hurdle. This brings to light a critical trade-off between the efficiency of sampling methods and their scalability to longer protein sequences.

Moreover, this paper identifies a crucial gap in the field: the lack of effective benchmarking methods for benchmarking the uniform random distribution for longer protein sequences. As we advance in our capability to simulate proteins of realistic lengths, the exponential increase in the number of possible conformations presents a formidable challenge in benchmarking these models and approaches. This gap underscores the need for innovative approaches in algorithm development and benchmarking, capable of managing the intricacies and complexities of real-sized protein folding.

In conclusion, our exploration into the realms of uniform randomness and (re)sampling techniques in the context of the HP-protein folding problem has revealed several key challenges and considerations. Despite the potential of stochastic break sampling in reducing the need for resampling compared to the deterministic break sampling method, its tendency to generate invalid conformations with colliding amino acids remains a significant limitation. This necessitates frequent resampling, a strategy that becomes increasingly impractical as protein lengths grow. On the other hand, recursive methods, both stochastic and heuristic, offer some respite but at the cost of running in non-polynomial time, thus presenting their own set of challenges.

8 FUTURE WORK

Future research should focus on addressing the challenges highlighted in this study. This includes developing more efficient resampling techniques, adapting sampling methods for longer protein sequences, and innovating in benchmarking methodologies. Additionally, further investigation into the mirroring bias observed in recursive sampling and exploring computational strategies could be interesting avenues to follow.

Another, more critical area for future research is the quantification of invalid conformations within the protein conformational landscape, vital for refining the valid solution space Sv as defined in formula (3). This requires developing algorithms to identify non-viable protein structures, considering overlaps and collisions. This future research would not only enhance our understanding of the protein folding space but also complement the findings related to the unique sample count Su in formula (4). By accurately enumerating these invalid conformations, future research could lead to more efficient sampling methods and advance the fields of protein folding.

9 ACKNOWLEDGEMENTS

I would like to express my gratitude to Daan van den Berg for his outstanding mentorship and guidance over the six months. Working closely with you has been a blend of challenging, insightful, and enjoyable experiences. Your exceptional guidance has not only elevated my performances to new heights, but also profoundly reshaped my view of what university should encompass — fostering a spirit of academic collaboration and personal growth. I eagerly look forward to our continued collaboration in future projects.

In the same vein, my appreciation extends to the members of our research group, whose support and collaborative efforts have been invaluable. Their insightful questions, constructive feedback, and engaging sparring have not only sparked inspiration but also played a crucial role in navigating through the complexities of this research.

REFERENCES

[1] Moein Atari and Nayereh Majd. 2022. 2D HP protein folding using quantum genetic algorithm. In *2022 27th International Computer Conference, Computer Society of Iran (CSICC)*. IEEE, 1–8.

[2] Bonnie Berger and Tom Leighton. 1998. Protein folding in the hydrophobic-hydrophilic (HP) is NP-complete. In *Proceedings of the second annual international conference on Computational molecular biology*. 30–39.

[3] Nabil Boumedine and Sadek Bouroubi. 2019. A new hybrid genetic algorithm for protein structure prediction on the 2D triangular lattice. *arXiv preprint arXiv:1907.04190* (2019).

[4] Thang N Bui and Gnanasekaran Sundarraj. 2005. An efficient genetic algorithm for predicting protein tertiary structures in the 2D HP model. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. 385–392.

[5] Nigel Chaffey. 2003. Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P. Molecular biology of the cell. 4th edn.

[6] Thomas E Creighton. 1988. The protein folding problem. *Science* 240, 4850 (1988), 267–267.

[7] Fábio L Custódio, Hélio JC Barbosa, and Laurent E Dardenne. 2004. Investigation of the three-dimensional lattice HP protein folding model using a genetic algorithm. *Genetics and Molecular Biology* 27 (2004), 611–615.

[8] Pedro A Diaz-Gomez and Dean F Hougen. 2007. Empirical Study: Initial Population Diversity and Genetic Algorithm Performance. *Artificial Intelligence and Pattern Recognition* 2007 (2007), 334–341.

[9] Ken A Dill and Justin L MacCallum. 2012. The protein-folding problem, 50 years on. *science* 338, 6110 (2012), 1042–1046.

[10] Ken A Dill, S Banu Ozkan, M Scott Shell, and Thomas R Weikl. 2008. The protein folding problem. *Annu. Rev. Biophys.* 37 (2008), 289–316.

[11] Christopher M Dobson. 1999. Protein misfolding, evolution and disease. *Trends in biochemical sciences* 24, 9 (1999), 329–332.

[12] Agoston E Eiben and James E Smith. 2015. *Introduction to evolutionary computing*. Springer.

[13] Mario Garza-Fabre, Eduardo Rodriguez-Tello, and Gregorio Toscano-Pulido. 2015. Constraint-handling through multi-objective optimization: the hydrophobic-polar model for protein structure prediction. *Computers & Operations Research* 53 (2015), 128–153.

[14] Ian P Gent and Toby Walsh. 1996. The TSP phase transition. *Artificial Intelligence* 88, 1-2 (1996), 349–358.

[15] William E Hart and Sorin Istrail. 1997. Robust proofs of NP-hardness for protein folding: general lattices and energy potentials. *Journal of Computational Biology* 4, 1 (1997), 1–22.

[16] Reitze Jansen, Ruben Horn, Okke van Eck, Sarah L Thomson, and Daan van den Berg. 2023. Can HP-protein folding be solved with genetic algorithms? Maybe not. (2023).

[17] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 7873 (2021), 583–589.

[18] Jesse Kommandeur. 2024. *Protein Folding: Uniform Random Sampling using the HP-Model*. https://github.com/jessekommandeur/Protein-Folding

[19] Jesse Kommandeur. 2024. *Protein Folding: Uniform Random Sampling using the HP-Model*. https://www.kaggle.com/datasets/jessekom/protein-folding-hp-model

[20] Kit Fun Lau and Ken A Dill. 1989. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules* 22, 10 (1989), 3986–3997.

[21] Cheolju Lee, Soon-Ho Park, Min-Youn Lee, and Myeong-Hee Yu. 2000. Regulation of protein function by native metastability. *Proceedings of the National Academy of Sciences* 97, 14 (2000), 7727–7731.

[22] Cheng-Jian Lin and Ming-Hua Hsieh. 2009. An efficient hybrid Taguchi-genetic algorithm for protein folding simulation. *Expert systems with applications* 36, 10 (2009), 12446–12453.

[23] Aaron Luntala Nsakanda, Wilson L Price, Moustapha Diaby, and Marc Gravel. 2007. Ensuring population diversity in genetic algorithms: A technical note with application to the cell formation problem. *European journal of operational research* 178, 2 (2007), 634–638.

[24] Arnold L Patton, William F Punch III, and Erik D Goodman. 1995. A Standard GA Approach to Native Protein Conformation Prediction.. In *ICGA*. 574–581.

[25] Max F Perutz, Michael G Rossmann, Ann F Cullis, Hilary Muirhead, Georg Will, and ACT North. 1960. Structure of hæmoglobin: a three-dimensional Fourier synthesis at 5.5-Å. resolution, obtained by X-ray analysis. *Nature* 185, 4711 (1960), 416–422.

[26] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. 2020. Improved protein structure prediction using potentials from deep learning. *Nature* 577, 7792 (2020), 706–710.

[27] Philip J Thomas, Bao-He Qu, and Peter L Pedersen. 1995. Defective protein folding as a basis of human disease. *Trends in biochemical sciences* 20, 11 (1995), 456–459.

[28] Ron Unger and John Moult. 1993. Genetic algorithms for protein folding simulations. *Journal of molecular biology* 231, 1 (1993), 75–81.
[29] Okke Van Eck and Daan Van Den Berg. 2023. Quantifying Instance Hardness of Protein Folding within the HP-model. In *2023 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 1–7.
[30] Shuihua Wang, Lenan Wu, Yuankai Huo, Xueyan Wu, Hainan Wang, and Yudong Zhang. 2016. Predict two-dimensional protein folding based on hydrophobic-polar lattice model and chaotic clonal genetic algorithm. In *Intelligent Data Engineering and Automated Learning*. Springer, 10–17.
[31] Jianzhi Zhang. 2000. Protein-length distributions for the three domains of life. *Trends in Genetics* 16, 3 (2000), 107–109.

Appendix A GITHUB AND KAGGLE

The source code of this Protein Folding research project is hosted on GitHub and Kaggle using the MIT License. Under the public *Protein Folding* repository, I have several code repositories:

(1) **Algorithms** - This Github folder contains all notebooks with algorithmic code for each of the five experiments. All code was written in python
(2) **Data** - This Github folder contains all generated datasets up to 25MB that were used in this research project. Larger datasets can be found on Kaggle.
(3) **Data** - This Kaggle page contains all generated datasets up to 10GB that were used in this research project. The Kaggle page has an overall usability score of 8.11.
(4) **Timing Data** - This Github folder contains the timing notebook and data that was used to time the experiments. All timing data is also included in the notebook itself.
(5) **Validation** - This Github folder contains all notebooks with validation statistics and code for each of the experiments. All statistics were based on the SciPy package.
(6) **Visualizations** - This Github folder contains all notebooks that were used to visualize the data of the experiments. It also contains a subfolder with a the full-size figures.
(7) **Thesis** - In this Github folder you can view and/or read the thesis itself. If you would like to share this paper, contact jesse.k@live.nl
(8) **Presentations** - This Github folder contains all presentations that were given during the weekly thesis sessions under the supervision of Daan van den Berg at the VU Campus in Amsterdam.

Appendix B DATASET GENERATION

Experiments 1 and 2 (Datasets 1HP200 and 2HP200) feature protein sequences with lengths varying from 5 to 200 amino acids, increasing in steps of 5. Each of these datasets encompasses 40,000 instances. Experiments 3 and 4 (Datasets 3HP100 and 4HP100) follow a similar structure, these experiments involve sequences ranging from 5 to 100 amino acids. The number of instances for each is set at 20,000. Experiment 5 (Dataset 5HP15) stands out due to its high number of instances, totalling 597,703, but it restricts the sequence lengths to just 5, 10, and 15 amino acids. A short overview of the different dataset is given in Table 3 below.

Table 3: Datasets overview

Experiment	*Dataset*	*Length*	*Instances*
1	1HP200	{5, 10, 15, ..., 200}	40.000
2	2HP200	{5, 10, 15, ..., 200}	40.000
3	3HP100	{5, 10, 15, ..., 100}	20.000
4	4HP100	{5, 10, 15, ..., 100}	20.000
5	5HP15	{5, 10, 15}	597.703

Appendix C TIMING DATA

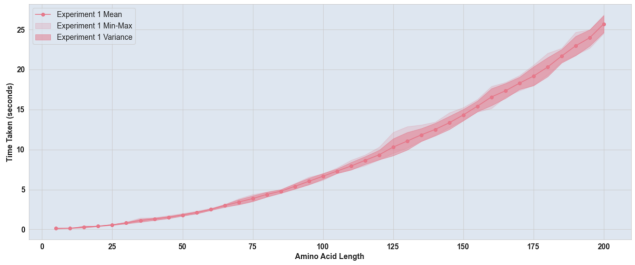


Figure 11: Time complexity experiment 1
Experiment 1's graph displays a polynomial time increase in sampling time with amino acid length, indicated by a line plot of the mean, variance (shaded area), and range (min-max shaded area).

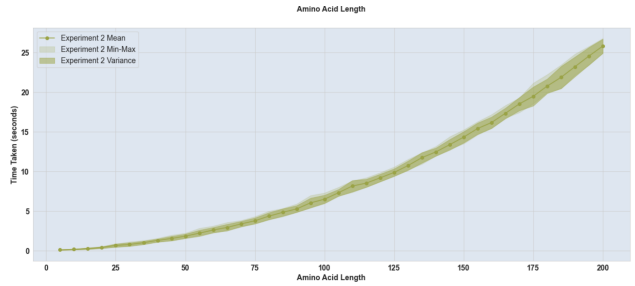


Figure 12: Time complexity experiment 2
Experiment 2's plot mirrors the polynomial pattern of Experiment 1, showing mean sampling times with variance and range as amino acid length extends.

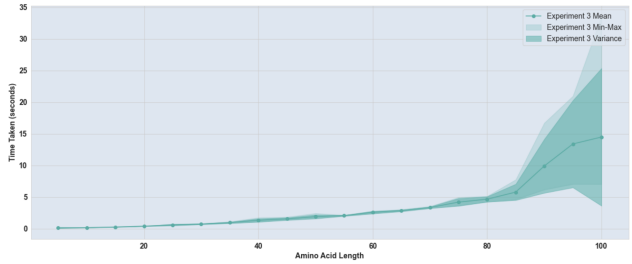


Figure 13: Time complexity experiment 3
Experiment 3 shows a distinctive non-polynomial time increase, with a sharper rise in mean time and a broad variance in the higher amino acid lengths.

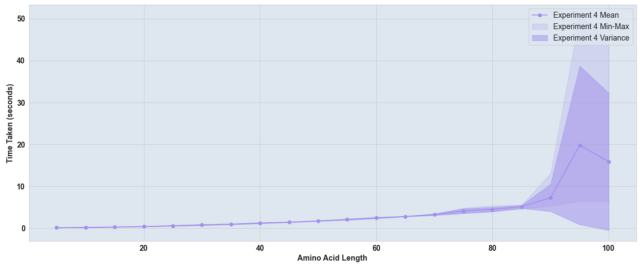


Figure 14: Time complexity experiment 4
Experiment 4's graph also reveals a non-polynomial growth in time, particularly notable for longer amino acid lengths, with a substantial increase in variance towards the end.

Appendix D EXPERIMENT VALIDATION

In this first section we present the results of the Analysis of Variance (ANOVA) tests conducted to assess the consistency of data across different protein lengths for experiment 1 and 2. The ANOVA tests were performed for each protein length category to determine if there were significant differences in the 'Amino Acids on Grid' metric across different experiments. The following table summarizes the F-values and p-values obtained for each protein length. The mean F-value and p-value across all lengths were calculated to provide an overview of the overall variance and significance. The null hypothesis (H0) for each ANOVA test assumes no significant difference in means across the datasets for each protein length.

Table 4: ANOVA Test Results Experiment 1

Amino Acid Length	F-Value	p-Value	Hypothesis
5	NaN	NaN	H0
10	0.185	0.947	H0
15	0.770	0.545	H0
20	0.913	0.455	H0
25	0.251	0.909	H0
30	1.088	0.361	H0
35	0.810	0.519	H0
40	0.513	0.726	H0
45	0.539	0.707	H0
50	1.263	0.282	H0

Table 4 and 5 summarize the ANOVA test results per amino acid length, facilitating a representation of the statistical analysis for experiment 1 and 2. The 'Hypothesis' column indicates the outcome of each test, with 'H0' signifying that the null hypothesis was not rejected for any of the protein lengths, implying no significant differences in the means across the datasets.

In experiments 3 and 4, we conducted an ANOVA test to examine the stability of recursion counts across different runs. This test compared the frequency of recursions relative to total placement attempts across five datasets in each experiment. The null hypothesis (H0) assumed there was no significant difference in recursion proportions across these datasets, while the alternative hypothesis (H1) indicated potential variations. The F-score and corresponding p-value for each dataset comparison were calculated to determine

Table 5: ANOVA Test Results Experiment 2

Amino Acid Length	F-Value	p-Value	Hypothesis
5	2.387	0.049	H1
10	1.545	0.186	H0
15	1.138	0.337	H0
20	0.976	0.419	H0
25	0.496	0.738	H0
30	0.996	0.408	H0
35	0.739	0.565	H0
40	0.112	0.978	H0
45	1.565	0.181	H0
50	1.271	0.279	H0

the statistical significance of any observed differences. A p-value below 0.05 was considered indicative of rejecting the null hypothesis. The results of this analysis are summarized in Table 3.

Table 6: ANOVA Test Results experiment 3

Amino Acid Length	F-Value	p-Value	Hypothesis
5	NaN	NaN	H0
10	0.519	0.722	H0
15	0.652	0.626	H0
20	2.409	0.047	H1
25	0.520	0.721	H0
30	0.479	0.751	H0
35	0.810	0.519	H0
40	0.891	0.468	H0
45	1.101	0.354	H0
50	0.787	0.533	H0

Table 7: ANOVA Test Results Experiment 4

Amino Acid Length	F-Value	p-Value	Hypothesis
5	NaN	NaN	H0
10	0.867	0.483	H0
15	0.524	0.718	H0
20	0.671	0.612	H0
25	0.832	0.505	H0
30	0.385	0.820	H0
35	1.095	0.357	H0
40	1.693	0.149	H0
45	1.342	0.252	H0
50	0.467	0.760	H0

Table 6 and 7 summarize the ANOVA test results per amino acid length, facilitating a representation of the statistical analysis focused on the stability of recursion counts across experiments 3 and 4. The test results indicated general consistency in recursion proportions across most amino acid lengths. Specifically, the analysis revealed no significant differences for lengths 10, 15, 25, 30, 35, 40, 45, and 50, as evidenced by their p-values being well above the 0.05 significance threshold. However, a notable exception was observed for Amino Acid Length 20, where the F-value of 2.409 and a p-value of 0.047 suggested significant variation, thereby rejecting the null hypothesis for this specific length. This implies that while the recursion behaviour remained stable for most lengths, some variability was detected at Length 20.