

Optimization-Based Prediction of Protein Structures in the 2D & 3D HP Model:

A COMPARATIVE STUDY OF METAHEURISTIC APPROACHES

KEYWORDS: PROTEIN FOLDING, 2D HP MODEL, METAHEURISTICS, OPTIMIZATION,
CONFORMATIONAL SEARCH , HYDROPHOBIC INTERACTIONS.

Authors: Yassin M. Alam Elden, Omar M. Hegab, Mariam E. Elshamy



Abstract:

The Hydrophobic-Polar (HP) model is a simplified lattice-based representation used to understand protein folding. In this work, we propose optimization-based strategies for predicting protein structures using the 2D HP model. Several metaheuristic algorithms [e.g., Genetic Algorithm (GA), Simulated Annealing (SA), Particle Swarm Optimization (PSO)] are evaluated for their effectiveness in predicting low-energy conformations. Experimental results on benchmark protein sequences demonstrate the performance, convergence behaviour, and structural diversity achieved by the proposed techniques. This study highlights the potential of optimization algorithms for protein structure prediction in reduced dimensionality models.

1. Introduction

Protein folding is the process where a protein twists and folds into a specific three-dimensional shape that allows it to do its job in the body. This starts as the protein is being built, with its amino acids interacting to create a final structure dictated by their sequence. Getting this shape right is super important because if the protein folds incorrectly, it can stop working or even become harmful. Misfolded proteins have been linked to serious diseases like Alzheimer's, Parkinson's, and cystic fibrosis. But protein folding isn't just about these diseases, it also plays a huge role in producing vital hormones like insulin.

Insulin is a small protein that needs to fold correctly to help regulate blood sugar. Diabetes, a condition affecting over 537 million people worldwide today, happens when the body can't produce or use insulin properly. Unfortunately, there's a growing global shortage of insulin — nearly half of the people who need it don't have reliable access. One big reason for this shortage is the challenge in manufacturing insulin at scale, since producing properly folded, functional insulin proteins is complex. That's why understanding protein folding better is so important, it could lead to more efficient ways to make insulin and help millions around the world.

To study protein folding, scientists use simplified models like the Hydrophobic-Polar (HP) lattice model. This model captures key features of how proteins fold while making the problem easier to handle. It helps researchers explore the many shapes a protein can take and understand how it settles into its most stable,

low-energy form often the shape it naturally adopts in the body.

But protein folding is an incredibly complex puzzle. Finding the perfect folded structure means searching through an enormous number of possibilities so many that exact methods, which try every option, quickly become impossible to use as proteins get bigger. They also struggle with “local minima,” which are shapes that seem good but aren't the absolute best.

That's why optimization algorithms are so useful. They are significantly quicker and still quite accurate since they can intelligently search through a large number of alternatives without verifying each one. These methods are adaptable to various protein sequences and folding difficulties because they may also be modified and improved.

In this paper, we focus on using these optimization methods with the HP lattice model to predict protein structures with low energy, meaning they're stable and likely biologically active. We'll compare different algorithms to see which performs best and how changing certain parameters affects the results. Our goal is to better understand protein folding and improve predictions a step that could ultimately help in designing proteins like insulin more efficiently, helping to address the urgent global shortage.

2. Literature Review

2.1 Overview of the 2D HP Model and Its Variants

The Hydrophobic–Polar (HP) model, introduced by Dill (1985, 1989), is one of the most widely studied lattice-based abstractions for the protein folding problem. In this simplified representation, amino acids are classified as either hydrophobic (H) or polar (P). Proteins are mapped onto lattice grids, where the folding process is modeled as a self-avoiding walk (SAW). The primary goal is to maximize the number of non-consecutive hydrophobic–hydrophobic (H–H) contacts, which correspond to stabilizing interactions in real proteins.

Several lattice variants have been explored to capture different folding dynamics:

- **2D square lattice**; the most common form, where residues can move in four cardinal directions.
- **2D triangular (hexagonal) lattice**; allowing six movement directions, which provides higher flexibility and sometimes more accurate folding predictions.
- **3D cubic lattice**; it requires significantly more computing power, the 3D cubic lattice is a logical extension that more accurately captures the intricacy of actual protein structures.

The HP model is still used as a standard for evaluating optimisation methods because it provides an equilibrium between biological relevance and tractability, even with its simplifications.

2.2 Survey of Previous Algorithms

2.2.1 Exact Methods: Dynamic Programming and Branch and Bound

Early approaches to solving the protein folding problem in the HP model relied on exact algorithms. Dynamic programming and branch-and-bound methods (Chen & Huang, 2005)

were employed to systematically search the conformational space. While these methods guarantee optimal solutions, their applicability is limited to very short sequences due to the exponential growth of the search space. For sequences longer than ~20–25 residues, exact methods become computationally infeasible.

2.2.2 Heuristics: Hill Climbing and Greedy Approaches

To address scalability, heuristic strategies such as hill climbing and greedy algorithms were developed. These methods construct solutions incrementally or iteratively improve a single conformation. Although faster than exact methods, they often suffer from getting trapped in local minima. As a result, their efficacy dramatically declines with larger proteins, where the energy landscape is extremely complex and multimodal.

2.2.3 Metaheuristics: GA, PSO, SA, ACO, DE, etc.

Metaheuristic algorithms have emerged as the most promising approaches for HP model protein folding.

- **Genetic Algorithms (GA)**: To preserve variety and prevent premature convergence, take advantage of population-based search, crossover, and mutation to investigate a large solution space.
- **Simulated Annealing (SA)**: The annealing process in metallurgy serves as a model for Simulated Annealing (SA), which probabilistically accepts subpar answers early in the search to assist avoid local minima.
- **Particle Swarm Optimization (PSO)**: Particles' social behaviour is modelled, and their placements are updated according to both global and personal best solutions.
- **Ant Colony Optimization (ACO)**: Exploits pheromone trails to guide folding paths, particularly effective for combinatorial representations.

- **Differential Evolution (DE):** Used less frequently for discrete protein folding problems, although it performs at continuous optimisation.

In 2D and 3D lattices, comparative studies (Su et al., 2011; Boumedine & Bouroubi, 2019) demonstrate that GA typically produces the most dependable results, however hybrid GA-based methods and SA also perform competitively.

2.3 Limitations in Previous Work

Although progress has been made, several limitations persist in existing literature:

- **Premature convergence:** Many algorithms, especially PSO and GA without diversity-preserving mechanisms, risk converging to suboptimal local minima.
- **Poor diversity maintenance:** Ensuring a wide range of conformational candidates is critical; otherwise, search spaces are insufficiently explored.
- **Scalability issues:** Algorithms that perform well on short benchmark sequences often fail to scale effectively to longer sequences or 3D lattices, where complexity grows exponentially.
- **Computational cost:** Methods that balance accuracy and runtime remain limited, particularly for real biological sequences beyond synthetic benchmarks.

These weaknesses show how reliable, scalable, and hybridised methods are required to manage 2D and 3D folding situations with precision and efficiency.

3. Methodology

3.1. Problem Representation

2D Square Lattice Description: In HP model proteins structures are depicted as 2D square lattice graphs. Each vertex of the graph corresponds to a position in the amino acid

sequence of the protein. It adheres to the following rules:

- **Lattice Grid** - Each of the amino acids may be either placed in the specific grid cell or the cell may be empty.
- **Mobility** - Each amino acid may move to adjacent cells in the four cardinal directions. The movement of the protein structure resembles the folding process as it attempts to find the optimal structure.

Encoding scheme: To represent the protein sequence and its corresponding structure, we can employ either relative or absolute encoding schemes:

1. **Absolute Encoding:** Each amino acid is represented by its fixed position on the lattice.

For example, a sequence of amino acids could be encoded as a series of coordinates, such as (x1, y1), (x2, y2).

2. **Relative Encoding:** This method encodes the movement of each amino acid relative to its predecessor. For instance, the first amino acid's position can be fixed at the origin (0, 0),

and subsequent positions can be defined by moves such as "up," "down," "left," or "right."

Energy function definition (hydrophobic contact model): The energy function plays an important role in determining the stability of a given protein conformation. In the case of the HP model, energy function focuses exclusively on hydrophobic interactions:

- **Hydrophobic-Hydrophobic (H-H) Contacts:** The energy function gives positive rewards

for arrangements that provide the maximum number of H-H contacts. Each H-H contact is given a Stability relevant negative energy value.

$$E = -\alpha \cdot (\text{Number of H-H contacts}) + \beta \cdot (\text{Penalties associated with invalid configurations})$$

Where:

- (α) alpha: α is a constant that describes how much H-H contacts contribute towards the energy of the system.
- Invalid (self-intersecting) configurations violate the physical reality of protein folding and so need to be penalized, and thus the penalty term is included.

4. Optimization Techniques

Three different optimization techniques are used to find protein conformations with the lowest amount of energy.

4.1. Genetic Algorithm (GA):

The Genetic Algorithm belongs to those metaheuristics, which were inspired by nature, or more specifically, the process of natural selection and genetics. It is particularly suitable for solving complex problems of optimization such as protein folding in the HP model.

The Genetic Algorithm step takes N different conformations of the same sequence of amino acids and by roulette wheel mechanism chooses 2 parents to recombine into one offspring. The probability $p(S_i)$ of a structure being selected as one of the two parents, so there are a feature for Genetic Algorithm (GA)

$$p(S_i) = \frac{E_i}{\sum_{j=1}^N E_j}$$

Key Features of Genetic Algorithm:

- 1. Population-Based Search:** The GA functions using a group of potential solutions (individuals), enabling it to investigate various areas of the solution space simultaneously.
- 2. Encoding:** Each individual is generally represented as a string (chromosome) that represents a solution. In the case of protein folding, this might be a series of movements on a lattice.

3. Fitness Evaluation: A fitness function evaluates how effectively each individual addresses the problem. In the context of protein folding, fitness is determined by the configuration's energy, taking into account hydrophobic interactions and penalties for invalid arrangements.

4. Selection: Various methods are used to select individuals for reproduction based on their fitness, such as:

- *Tournament Selection:* A random group of individuals is selected, and the best individual is chosen.
- *Roulette Wheel Selection:* Participants are chosen based on their fitness ratings.

5. Crossover: In order to produce offspring, crossover combines segments from two parent individuals. This introduces new genetic variations into the population and is crucial for investigating different solutions. Multi-point Crossover; This allows for more intricate combinations of parental traits by selecting multiple crossover points.

6. Mutation: By introducing random changes to individuals, this step preserves population diversity and helps avoid early stagnation. For instance, it could entail changing a protein folding sequence move.

7. Elitism: To ensure high quality, the top performers from the present generation are retained for the following generation.

8. Termination Criteria: The algorithm typically runs for a fixed number of generations or until a convergence criterion is met, such as a lack of significant improvement in fitness.

3.1.1. Fitness Function

In the (GA) Genetic Algorithm, the fitness function is essential for assessing the caliber of protein conformations. Within the framework of the HP model, it comprises two principal elements:

1. Total Number of H-H Contacts:

- Total Number of H-H Contacts: The primary measure of a protein's stability is

the total number of hydrophobic-hydrophobic (H-H) contacts.

- Configurations that optimize these interactions are desirable, as each H-H contact has a negative impact on the energy score.
- The energy function can be defined as:

$$E = - (\text{Number of H-H contacts})$$

2. Penalties for Invalid Configurations:

- To guarantee realistic folding, invalid configurations are penalized, such as self-intersections, in which the protein overlaps itself.
- For every invalid configuration, a penalty term is added to the energy score.

$$E_{\text{total}} = - (\text{Number of H-H contacts}) + \beta \cdot (\text{Penalties for invalid configurations})$$

- (**β**) **beta**: it is a constant that determines the severity of the penalties applied.

3.1.2. Implementation Details

Initialization Strategy: The population of potential solutions is initialized randomly. In order to replicate the folding process, each member of the population represents a random series of movements. A set of integers that represent the movement directions on the lattice, for instance, can be used to represent the sequence.

Parameters and Tuning: Population Size: Usually set at 100–300 people. A larger population increases diversity but also computational cost, but the population used is 200. Mutation Rate: Usually set between 0.02 and 0.2 points. Balanced between exploration and exploitation, this rate establishes the probability of random changes in the progeny. Although the code does not specifically define it, a high crossover rate (e.g. A. 0.7 to 0.9) is typically favoured in order to promote genetic mixing.

Stopping Criteria: Maximum Iterations: To guarantee a limit on computation time, the algorithm executes for a predetermined number

of generations, or 5000 iterations. Convergence Threshold:

If, after a predetermined number of generations, the best fitness score does not considerably improve (e.g. G. 100), the algorithm may stop early, signifying that a workable solution has been found.

4.2. simulated annealing (SA):

The Simulated Annealing algorithm is a type of probabilistic metaheuristic that is based on the physical process of annealing in metallurgy. In this process, a material is heated and then slowly cooled to get rid of flaws and create a stable crystalline structure. It is particularly effective for tackling complex optimization problems such as protein folding in the HP model.

The SA algorithm looks for solutions by moving from one conformation to another over and over again. Moves that lower the system's energy are allowed, but moves that raise energy may still be allowed, but the chance of this happening goes down over time (controlled by a temperature parameter). This helps the algorithm get away from local minima and get closer to the best solution overall. The acceptance probability of a move that changes the energy by ΔE at temperature T is given by:

$$p = \exp\left(-\frac{\Delta E}{T}\right)$$

Key Features of simulated annealing:

1- Single Initial solution

Starting point for the search, begins with a randomly chosen or heuristically derived

2- Encoding

Like other HP model approaches, the protein conformation is encoded as a series of movements on a lattice.

3- fitness evaluation

conformation's hydrophobic-interactions, penalties for invalid arrangements, and general stability all affect its fitness (or energy).

4- move generation

The current conformation is subjected to minor, random modifications (such as rotating a segment or moving a portion of the chain).

5- acceptance rule

- if the conformation's energy is lower, it is accepted
- if the conformation's energy is higher, it is accepted with probability.

6- cooling schedule

Over time, the acceptance of worse solutions decreases as the temperature T is progressively lowered in accordance with a cooling schedule (e.g., geometric $T \leftarrow \alpha T$ with $\alpha < 1$).

7- termination criteria

When the temperature drops to a very low level and no improvement is seen, or after a predetermined number of iterations, the algorithm terminates.

4.2.1. Fitness Function

In the simulated annealing technique, the quality of a particular protein conformation on a 2D square lattice is determined by the fitness function. The Hydrophobic-Polar (HP) model serves as the foundation for the implemented function, which maximizes hydrophobic stability. It is based on the following two points:

1. Total Number of H-H Contacts:

Counts the number of non-consecutive H-H contacts after folding the protein chain, more H-H contacts indicate stronger hydrophobic core formation and higher structural stability.

Implementation technique:

- The function examines each of the four adjacent lattice positions (up, down, left, and right) for every hydrophobic residue (H).
- The energy score is lowered by one if a neighbor is also a H and is not directly connected in the sequence (non-consecutive).
- Division by 2 at the end avoids double-counting contacts

(Formula used: $E = -\frac{1}{2} \times N_{H-H}$)

2. Self-Avoiding Constraint

- In order to prevent the chain from overlapping itself, the folding coordinates are generated (Self-Avoiding Walk, SAW).
- The SA process completely avoids overlapping conformations.

4.2.2. Implementation Details

Initialization Strategy: For a given protein sequence on a 2D square lattice, the algorithm first generates a random self-avoiding walk (SAW). This is accomplished by choosing the movement directions (left, right, down, and up) at random while making sure that no lattice position is visited more than once. The initial conformation from which the search process begins is formed by the resulting coordinates.

Parameters and tuning:

- Initial Temperature (T_0): Regulates the likelihood that less-than-ideal solutions will be accepted initially. It is set to 1.0 in the code, which increases the possibility of initially attempting less-than-ideal moves.
- Cooling Rate (α): Establishes how quickly the temperature drops following each iteration. In order to balance exploration and exploitation, a value of 0.95 is employed, which offers a gradual cooling schedule.
- Iterations per Run: The SA algorithm generates, assesses, and rejects possible moves in 5000 iterations per run.
- Number of Runs: To lessen the impact of chance and raise the likelihood of discovering a nearly ideal fold, the algorithm is run 50 times. The best outcome from each run is noted.

Stopping criteria:

- Iteration Limit: Regardless of whether progress is still being made, each run ends after a predetermined number of iterations (5000).
- Temperature Limit: The search process is essentially frozen as the cooling

schedule goes on because the temperature eventually drops to such a low level that it is uncommon to accept worse solutions.

- **Best-Energy Tracking:** The algorithm keeps the conformation with the lowest energy across all runs. The search can end early if a known optimal energy (based on sequence length) is reached.

4.3. Particle Swarm Optimization (PSO):

Particle swarm optimization (PSO) is a computational technique used in computational science that iteratively attempts to enhance a candidate solution in relation to a specified quality metric in order to optimize a problem. It moves a population of potential solutions – known as particles- around the search space using simple mathematical formulae that control the particle's position and velocity in order to solve a problem. In addition to being guided towards the best-known places in the search-space, which are updated as other particles discover better positions, each particle's movement is impacted by its local best-known position. This is expected to move the swarm toward the best solutions.

Key features of PSO:

1. Swarm-Based search: PSO works using a population of particles, where each particle is a candidate solution (protein folding path). PSO relies on updating positions and velocities based on experience and collaboration instead of developing through crossover/mutation like GA.

2. Encoding: Each particle's position encodes a potential folding configuration (such as a sequence of lattice moves), the velocity shows how the folding configuration should differ in the following iteration.

3. Fitness evaluation: A fitness function evaluates the quality of each particle's position (candidate folding configuration). In the context of protein folding, the energy of the folded structure is usually used to determine fitness, where lower energy denotes a more stable protein. The calculation takes into

account the hydrophobic-hydrophobic interactions that should be maximized, while applying penalties for incorrect or overlapping conformations. Each particle continuously modifies its position to lower this energy, using both its own best-known solution and the swarm's best solution as guidance.

4. Personal and Global best: Each particle records pBest (personal best) which is the best solution the particle has found so far, and gBest (global best) which is the best solution found by the whole swarm. These two guide the particles toward better positions in the search space.

5. Velocity and Position Update: The particle movement is controlled by three main components: inertia (w), which sustains momentum and keeps the particle moving in the same direction; the cognitive component ($c1$), which pulls the particle toward its own personal best solution; and the social component ($c2$), which draws the particle toward the global best solution found by the entire swarm. Together, these factors ensure the balancing between exploration, where particles search new regions in the solution space, and exploitation, where the particles refine and enhance the already promising folding configurations.

6. Stochastic influence: Random numbers ($r1$, $r2$) are added in the velocity update to provide diversity in particle movement. This randomness lowers the probability of the swarm getting stuck in local optima. By the addition of controlled unpredictability, particles are more motivated to search through various folding configurations resulting in improving the search for the global minimum energy.

7. No Genetic Operators: PSO produces novel solutions without relying on evolutionary operators like crossover or mutation, unlike Genetic Algorithms (GA). The particle's motions, which are impacted by inertia, personal best, global best, and random variables, instead provide the means of exploration of the search space. Diversity is naturally maintained because each particle follows a distinct path determined by its velocity and experiences. This makes PSO fast

and easy to implement, while still preserving enough swarm variety to prevent premature convergence.

8. Elitism (Implicit): The global best solution is always tracked and maintained implicitly to make sure that the best-known fold is never lost.

9. Termination Criteria: PSO usually runs until either a maximum number of iterations is reached or the energy converges where time passes and the energy value is almost the same.

5. Experimental Setup

A) Datasets

For evaluating the performance of the Genetic Algorithm in protein folding using the HP model, specific benchmark protein sequences are utilized. These sequences are selected from established datasets to ensure comparability and reliability of results.

ID Seq	No. of AA	Sequences
1	20	HPHP PHHP HPPH PHHP PHPH
2	24	HHPP HPPH PPHP PHPP HPPH PPHH
3	25	PPHP PHHP PPPH HPPP PHHP PPPH H
4	36	PPPH HPPH HPPP PPHH HHHH HPPH HPPP PHHP PHPP
5	48	PPHP PHHP PHHP PPPH HHHH HHHH HHPP PPPH HHPP HHPP HPPH HHHH
6	50	HHPH PHPH PHHH HPHP PPHP PPHP PPPH PPPH PPHH HPHP HPHP HH
7	60	PPHH HPHH HHHH HHPP PHHH HHHH HHHP HPPP HHHH HHHH HHHH PPPH HHHH HHPH HPHP
8	64	HHHH HHHH HHHH PHPH PPHH PPHH PPHP PHHP PHHP PHPP HHPP HHPP HPHP HHHH HHHH HHHH
9	85	HHHH PPPH HHHH HHHH HHHH PPPH PPHH HHHH HHHH HHPP PPHH HHHH HHHH HPPP HHHH HHHH HHHH PPPH PPHH PPHH PPHH

Table 3.1. Dataset for Converted HP Sequence

B) Evaluation Metrics

Some important evaluation metrics are used to evaluate the Genetic Algorithm's performance in protein folding:

1. Minimum Energy Obtained: This measure shows the algorithm's lowest energy score during the optimization phase. A more stable protein conformation, indicating successful folding, is indicated by a lower energy value.

2. Number of H-H Contacts: Hydrophobic-hydrophobic (H-H) contact counts are added. The stability of the protein structure is directly correlated with this metric, making it essential.

A better arrangement of hydrophobic residues is suggested by a greater number of H-H contacts.

3. Computational Time: The total time taken for the Genetic Algorithm to complete the optimization process is measured. This covers time for evaluations and all generations. Effective algorithms should produce high-quality results with the least amount of computation time.

4. Success Rate Over Multiple Runs: The Genetic Algorithm is run several times to determine the success rate (e.g. 100 runs) using

the same parameters and protein sequence. The percentage of runs that achieve the optimal or near-optimal energy configuration is calculated. This metric indicates the reliability and robustness of the algorithm in finding good solutions.

C) Hardware and Software

A PC with a 12th Gen Intel(R) Core (TM) i7-12700 CPU operating at 2.10 GHz and 32 GB

of RAM was used to evaluate the Python implementation of the method. Nine benchmark sequences with a range of 20 to 85 amino acids were used for the experiments (see Table 3.1). Unger (1993) [9] provided the first eight sequences, while Huang (2010) [16] provided the ninth. The sequence ID, length, and matching HP string are shown in Table 3.1.

6. Results and Discussion

6.1. Tabulated results comparing methods:

6.1.1. HP sequences (data set in table 3.1) results

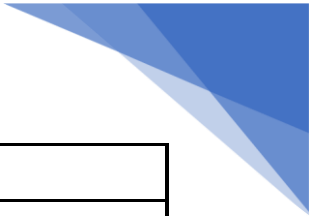
A) 2D lattice:

Protein ID	Optimal Energy (2D square)	GA				PSO		SA	
		2D square		2D triangular		Min energy	runtime (min)	Min energy	runtime (min)
		Min energy	runtime (min)	Min energy	runtime (min)				
1	-9	-9	0.31	-4	2.47	-9	0.90	-9	0.04
2	-9	-9	0.36	-4	2.59	-8	1.08	-8	0.09
3	-8	-8	0.34	-5	1.66	-7	1.05	-7	0.09
4	-14	-13	0.45	-8	1.8	-10	1.68	-11	4.09
5	-23	-19	0.62	-11	2.48	-18	1.95	-17	760.4
6	-21	-20	0.61	-11	4.73	-17	1.80		
7	-36	-30	0.73	24	3.70	-25	2.71		
8	-42	-34	0.80	-21	3.62	-32	2.85		
9	-53	-38	0.88	-35	4.85	-32	4.11		

Table 6.1 Comparison of GA, PSO, and SA in Protein Folding on 2D Lattice (HP Model)

B) 3D lattice:

Protein ID	Optimal energy (3D square)	GA Min Energy	GA Runtime (min)
1	-11	-11	0.98
2	-13	-11	1.27
3	-9	-9	1.44
4	-18	-16	4.63
5	-31	-25	43.08
6	-34	-25	104.36



7	-55		
8	-59		
9	-		

Table 6.2 Comparison of GA-obtained minimum energies with optimal benchmark energies for protein sequences on the 3D square lattice.

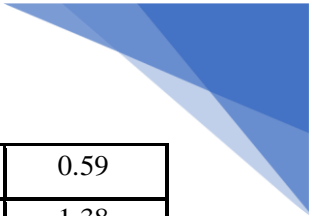
6.1.2. Converted FASTA sequences

Protein name	HP sequence	Length
1A7F_1	HHHHPPPPPPHPPHPPHPPPP	21
1A7F_2	HHPPPHRPHPPHHRPHHRPHRPPHHHPPHP	29
1APH_1	HHHHPPPPHPPHPPHPPPPPP	21
1APH_2	HHPPPHRPHPPHHRPHHRPHRPPHHHPPHPH	30
1ANP_1	PHPPPPPHRPHRPPHPPHPPHHHPPPHPP	28
1B17_1	HHHHPPPPPPHPPHPPHPPPP	21
1B17_2	HHPPPHRPHPPHHRPHHRPHRPPHHHPPHPH	30
1BZV_1	HHHHPPPPPPHPPHPPHPPPP	21
1BZV_2	HHPPPHRPHPPHHRPHHRPHRPPHHHH	26
1CKW_1	HHHRHPPPHHHHHPPPPPPPPHHRPH	25
1CKX_1	HHHRHPPPHHHHHPPPPPPPPHHRPH	26
1DOR_1	PHRPHRPPPHPPPHRPHRPPHHHHHHHPH	30
2BN3_1	HHHHPPPPPPHPPHPPHPPPP	21
2BN3_2	HHPPPHRPHPPHHRPHHRPHRPPHHHPPHPH	30
2KJI_1	HPPPHRPHPPHHRPHHHHHHRPHRPPHPPHPPHPPHPPPPHPPHPPHPPH	50
2MX4_1	HPPPHHHPPHHRPHHPPPPPPHHRPHRPPHHRPHHPPPPHHP	45
1BW5_1	HPPPHRPHRPPHPPPHPPPHPPPHRPHRPPHHHPPPHRPHRPPHPPHPPH HPPPPPPPPPPHHP	66
1GDC_1	HRHHPPPPHPPPHHHPPHPPHHHPPHHRPPPPHPPHPPPPHPPHPPHPP PPPHHPPPPPPHPPHHRPHRPH	72

Table 6.3 Dataset of converted FASTA sequences with their corresponding HP representations and sequence lengths used in the experiments.

Converted FASTA sequences results:

Protein name	GA					
	2D square		2D triangular		3D square	
	Minimum energy	Runtime (min)	Minimum energy	Runtime (min)	Minimum energy	Runtime (min)



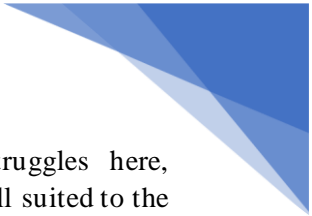
1A7F_1	-4	0.20	-3	2.00	-5	0.59
1A7F_2	-11	0.90	-7	3.23	-16	1.38
1APH_1	-6	0.61	-3	2.19	-7	0.61
1APH_2	-13	1.76	-8	3.52	-16	1.54
1ANP_1	-6	1.00	-5	2.91	-8	1.12
1B17_1	-4	0.57	-2	2.07	-5	0.60
1B17_2	-12	1.16	-8	4.70	-16	1.52
1BZV_1	-4	0.57	-2	3.32	-5	0.60
1BZV_2	-12	1.04	-8	4.69	-14	1.14
1CKW_1	-7	0.98	-6	3.05	-9	0.91
1CKX_1	-8	0.90	-6	3.01	-10	1.00
1DOR_1	-10	0.97	-6	3.64	-13	1.50
2BN3_1	-4	0.54	-3	2.11	-5	0.58
2BN3_2	-13	0.98	-7	3.67	-17	1.50
2KJI_1	-14	1.11	-10	5.34	-20	43.1
2MX4_1	-14	1.19	-11	5.30	-21	16.9
1BW5_1	-17	1.39	-15	10.00		
1GDC_1	-17	1.93	-15	7.55		

Table 6.3: GA results for converted FASTA sequences across 2D square, 2D triangular, and 3D square lattices, showing minimum energies and runtimes.

6.2.Discussion

The three algorithms clearly act very differently when trying to solve the protein folding problem, based on the results. The Genetic Algorithm (GA) was the best overall. It usually found the folds with the least energy and stayed very close to the best-known solutions. For the shorter benchmark sequences, GA consistently reached the exact optimal energy in less than a second. This shows how reliable it is even when it doesn't run for long. GA still did a good job as the sequences got longer, making folds that were almost perfect (for example, -20 instead of the -21 optimum in sequence 6). Simulated Annealing (SA) found good solutions, but they were usually not the best ones. Particle Swarm Optimization (PSO) had the most trouble, especially with longer chains. Another important consideration is runtime. In addition

to providing the best answers, GA completed the benchmark tests in less than a second on average. SA still didn't always reach the best folds and required more time (roughly 1-4 seconds). PSO was the slowest—it could take more than five seconds at times—and produced the least ideal results. This demonstrates how GA has an advantage in the HP model due to its capacity to simultaneously search a large number of possibilities and retain the best answers through elitism. The same pattern persisted when we switched to more realistic FASTA sequences. Even for longer proteins like 1BW5_1 and 1GDC_1, where its optimal energy was -17, GA continued to produce stable, low-energy folds. Even for these larger cases, the results remained reasonable (roughly 1–10 minutes), although the runtimes naturally



increased with sequence length and dimensionality. This implies that when applied to actual protein sequences, GA can scale up more effectively than SA and PSO. GA's population-based search appears to be the primary factor behind its success. It avoids becoming stuck in local minima by working with multiple candidate solutions simultaneously. Due to its probabilistic acceptance of worse moves, SA can occasionally avoid local traps as well. However, how well the cooling schedule is adjusted greatly affects how well SA performs.

7. Conclusion:

In this study, we investigated the protein folding problem using the Hydrophobic-Polar (HP) lattice model in both 2D and 3D, applying three metaheuristic optimization algorithms: Genetic Algorithm (GA), Simulated Annealing (SA), and Particle Swarm Optimization (PSO). The results on benchmark sequences and converted FASTA sequences highlight clear differences in algorithmic performance.

GA consistently outperformed the other methods across both 2D and 3D lattices. For short and medium-length sequences in the 2D square and triangular lattices, GA frequently reached the known optimal energies within very short runtimes, demonstrating its robustness and efficiency. SA achieved reasonable solutions and occasionally approached optimal energies, but its success was highly dependent on cooling schedule parameters and generally required more runtime. PSO, although conceptually powerful for continuous search spaces, showed weaker performance in the discrete HP lattice model, struggling particularly with longer sequences.

8. References

1. Chen, M., & Huang, W. (2005). A branch and bound algorithm for the protein folding problem in the HP lattice model. *Genomics Proteomics & Bioinformatics*, 3(4), 225–230.
2. Nanda Dulal Jana, Jaya Sil, Swagatam Das, Selection of appropriate metaheuristic algorithms for protein

PSO, on the other hand, struggles here, probably because it is not as well suited to the discrete lattice structure of the HP model and is intended for continuous spaces. Out of the three approaches, GA turned out to be the most dependable and effective overall. PSO proved to be the least successful in this situation, while SA showed promise but was less reliable. These results emphasize the significance of matching the algorithm to the problem structure, and for protein folding in the HP model, GA provides the best trade-off between speed, accuracy, and scalability.

Extending the experiments to 3D confirmed the increasing complexity of the conformational search space. GA was able to match or approximate optimal energies for shorter sequences but required substantially longer runtimes for larger proteins. For the longest benchmark sequences, results remained incomplete, underscoring the computational intensity of 3D folding. Nonetheless, GA showed better scalability than the other algorithms, making it the most promising approach among those tested.

Overall, this comparative study demonstrates that GA provides the best trade-off between accuracy, robustness, and runtime efficiency for protein structure prediction in the HP model. Future work will focus on hybrid approaches (e.g., GA combined with local search), parallelization using GPU acceleration, and extending the HP model with more realistic amino acid interaction potentials. Such advances could significantly improve the predictive power of optimization-based methods and bring them closer to applications in protein design and therapeutic research.

[https://doi.org/10.1016/s1672-0229\(05\)03031-7](https://doi.org/10.1016/s1672-0229(05)03031-7)

- structure prediction in AB off-lattice model: a perspective from fitness landscape analysis, *Information Sciences*, Volumes 391–392, 2017, Pages 28-64, ISSN 0020-0255, <https://doi.org/10.1016/j.ins.2017.01.020>.
3. Traykov, M., Department of Electrical Engineering, Electronics and Automatics, University Center for Advanced Bioinformatics Research, Yanev, N., Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Mavrevski, R., Department of Electrical Engineering, Electronics and Automatics, University Center for Advanced Bioinformatics Research, South-West University “Neofit Rilski,” Yurukov, B., & Department of Informatics, South-West University “Neofit Rilski.” (2018). Algorithm for protein folding problem in 3D lattice HP model. In *International Journal of Biology and Biomedicine* (Vol. 3).
 4. Su, S. C., Lin, C. J., & Ting, C. K. (2011). An effective hybrid of hill climbing and genetic algorithm for 2D triangular protein structure prediction. *Proteome science*, 9(Suppl 1), S19.
 5. Baynes, J.W.; Dominiczak, M.H. *Medical Biochemistry FIFTH EDITION*; Elsevier, 2019. fifth edition.
 6. Rashid, M.; Newton, M.A.H.; Hoque, M.; Sattar, A. Mixing Energy Models in Genetic Algorithms for On-Lattice Protein Structure Prediction. *BioMed research international* 2013, 2013, 924137. <https://doi.org/10.1155/2013/924137>
 7. Boumedine, N., & Bouroubi, S. (2019). A new hybrid genetic algorithm for protein structure prediction on the 2D triangular lattice. *arXiv preprint arXiv:1907.04190*.
 8. Dill, K. A. (1985). Theory for the folding and stability of globular proteins. *Biochemistry*, 24(6), 1501–1509. <https://doi.org/10.1021/bi00332a008>
 9. Lau, K. F., & Dill, K. A. (1989). A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22(10), 3986–3997. <https://doi.org/10.1021/ma00198a009>
 10. Sequence based features extraction. (n.d.). https://www.iitm.ac.in/bioinfo/SBFE/Amino_acid_prop_130.html
 11. Guo, Y., Tao, F., Wu, Z., & Wang, Y. (2017). Hybrid method to solve HP model on 3D lattice and to probe protein stability upon amino acid mutations. *BMC Systems Biology*, 11(S4). <https://doi.org/10.1186/s12918-017-0459-4>
 12. Dubey, S. P., Balaji, S., Kini, N. G., & Kumar, M. S. (2018). A novel framework for AB initio coarse protein structure prediction. *Advances in Bioinformatics*, 2018, 1–17. <https://doi.org/10.1155/2018/7607384>
 13. Bank, R. P. D. (n.d.). *RCSB PDB*. <https://www.rcsb.org/search>
 14. Li, H., Helling, R., Tang, C., & Wingreen, N. (1996). Emergence of preferred structures in a simple model of protein folding. *Science*, 273(5275), 666-669.

