Laboratoire d'Informatique Fondamentale, de Recherche Opérationnelle, de Combinatoire et d'Économétrie

Availaible on line at https://liforce.usthb.dz
bulletin-liforce@usthb.dz
Bulletin du Laboratoire 02 (2026) 01 - 24

Protein Folding Simulation Based on a Hybrid Approach Combining Genetic Algorithm and Improved Tabu Search

Nabil BOUMEDINE[1], Sadek BOUROUBI[2*]

[1]L'IFORCE Laboratory, Faculty of Science,
University of Boumerdes,
Independence Avenue, 35000 Boumerdes, Algeria.
[2]L'IFORCE Laboratory, Faculty of Mathematics,
University of Sciences and Technology Houari Boumediene,
P.B. 32 El-Alia, 16111, Bab Ezzouar, Algiers, Algeria

nboumedine@usthb.dz, bouroubis@gmail.com

Abstract : A proteins biological function is inherently tied to its three-dimensional structure, known as its native conformation. Although researchers have established the relationship between a proteins structure and its function, the precise mechanism that drives a protein to fold into its most stable form is still not fully understood. In biological terms, the optimal conformation corresponds to the structure with the lowest free energy. This understanding has led to the formulation of the protein folding problem (PFP) as a combinatorial optimization (CO) problem, where the goal is to predict a proteins tertiary structure. Over the years, numerous optimization methods have been proposed to address the PFP across various simplified models. In this paper, we develop an efficient hybrid genetic algorithm to solve the PFP and evaluate its performance against state-of-the-art algorithms using benchmark datasets.

Keywords: Combinatorial Optimization; Protein Folding Problem (PFP); Metaheuristics; Genetic Algorithm; Free Energy Minimization.

2020 Mathematics Subject Classification. Primary 90C27; Secondary 90C59.

Received : June 12, 2026

*Corresponding author: Nabil BOUMEDINE

1 Introduction

The protein has a fundamental role in all biological functions, as each of these functions is performed by a specific protein that scientists refer to as the native structure [32, 10]. According to Anfinsen's experiments, all information that allows the protein to fold into its native state is contained in its primary structure [9, 2]. Understanding the folding process and its mechanisms is the first step to addressing many serious diseases caused by aggregation or misfolding, such as Alzheimers, prion diseases, and mad cow disease [23, 5, 6, 25]. Experimental methods such as X-ray crystallography (XC) [13] and nuclear magnetic resonance (NMR) [34] have been used to determine the structure of real proteins. However, these methods are very costly in terms of equipment and time. Therefore, researchers widely study this problem using computational methods because of their speed and the quality of the solutions they provide. The native state of a protein is achieved when the chains that make up the protein fold into a minimum free-energy conformation [1]. Moreover, the folding process is driven primarily by hydrophobic interactions between amino acids, which are essential for the development of the native conformation. Each occurrence of these contacts reduces the total energy of the conformation by one unit [33]. Therefore, searching for an optimal conformation of a protein amounts to finding a structure that maximizes the number of hydrophobic contacts, known as H–H contacts. Although many simplified models have been introduced since the discovery of the relationship between protein function and structure to reduce the complexity of the protein folding problem (PFP), predicting protein structure remains one of the most challenging problems in biology, chemistry, and computer science [30]. The Hydrophobic-Polar (H–P) model, referred to as the H–P lattice model, is one of the most widely used models in the study of PFP [22]. The H–P model is based on a crucial principle in the folding process: the native state of proteins is largely determined by hydrophobic interactions among hydrophobic amino acids, which together form the core of the structure. In this model, the quality of a folding (i.e., the energy function E) for a given structure is measured by the number of hydrophobic contacts present [16]. Based on the H-P model, the PFP can be formulated as a combinatorial optimization problem using a lattice to represent the structure, where the goal is to identify the optimal conformation c^* that maximizes the number of H-H contacts in the conformational space C [19]:

$$c^* = \arg\min_{c \in C} E(c).$$

In this work, we present an efficient hybrid algorithm (HGAETS) that integrates a genetic algorithm (GA) with an enhanced Tabu search using the Lévy Flight Perturbations (ETS) approach to solve the H-P protein folding problem using a 2D square lattice model. The remainder of this paper is organized as follows. In Section 2, we review the related work on protein folding models and computational approaches. Section 3 presents the H-P simplified model for the protein folding problem, including its objective function, constraints, and solution representation. Section 4 describes the proposed algorithm in detail. In Section 5, we discuss the results and compare the performance of our algorithm with state-of-the-art methods. Finally, Section 6 summarizes the present work and discusses potential directions for future research.

2 Related works

Since the development of the H-P model, various heuristic and metaheuristic algorithms have been designed to solve the protein folding problem (PFP) on different lattice structures. In this paper, we focus on solving the PFP using the H-P model on a 2D square lattice. Based on this model, Unger and Moult [31] proposed a genetic algorithm (GA) that employs genetic operators (i.e., selection, crossover, and mutation) to explore the search space. The mutation operator introduces diversity, while the crossover operator generates new solutions by exchanging sub-sequences between two selected solutions. In the proposed GA, the probability of selecting a solution is proportional to its fitness and determined by the roulette selection operator. Later, several improved versions of the GA were introduced in [8, 11, 15, 26].

The Ant Colony Optimization (ACO) algorithm was first applied to the PFP in the 2D H-P square lattice model in [27], with an extended version presented in [28]. Particle Swarm Optimization (PSO) was also developed and successfully used to generate high-quality solutions on benchmark datasets [3]. Additionally, the Immune Algorithm (IA) has been applied to the PFP [7]. Several hybrid methods have also been proposed. For instance, a simple GA was combined with a tabu search (TS) algorithm [19], and the experimental results demonstrated that this hybrid approach outperforms the standard GA. In [29], the authors proposed a high-performance Genetic Algorithm (called ACGA) for the protein folding problem that allows all conformations, including non-self-avoiding walks, to appear during the evolutionary process. This strategy increases the probability of reaching low-energy native conformations with improved topological properties. Furthermore, the algorithm incorporates specialized translation-based crossover and mutation operators to enhance the search efficiency and solution quality. In [17], Zhang et al, proposed FoldingZero, a reinforcement learning-based approach for the HP protein folding problem. The algorithm integrates a two-head deep neural network with a regularized Monte Carlo Tree Search mechanism to guide the folding process. Unlike traditional metaheuristics, FoldingZero learns folding policies directly from experience through iterative self-improvement. The neural network evaluates intermediate conformations and predicts promising moves, while R-UCT balances exploration and exploitation during the search. The method achieved results comparable to several state-of-the-art approaches on standard HP benchmarks. In a recent study [35], Yang et al. applied Deep Reinforcement Learning to the 2D HP protein folding problem. Their approach combines a Deep Q-Network with an LSTM architecture to model the folding process as a sequential decision-making task. The LSTM network captures long-range interactions between amino acids, while reinforcement learning guides the search toward low-energy conformations. The method relies on experience replay and ϵ-greedy exploration to improve learning efficiency. Results on standard HP benchmarks demonstrated its ability to reach best-known solutions and outperform several previous reinforcement learning approaches.

Some approximation algorithms have also been developed for the PFP. Hart and Istrail proposed an algorithm achieving an approximation ratio of 1/4 (25%) for the 2D square lattice and 3/8 (38%) for the 3D cubic lattice [12]. Newman later improved this ratio to 1/3 (33%) for the 2D square lattice and 6/11 (54%) for the 3D cubic lattice [24].

3 Protein Folding in the HP Lattice Model

In the H-P simplified lattice model, all twenty amino acids are classified into two groups: the hydrophobic group (H) and the polar group (P)[27]. Given a protein sequence of n amino acids, this simplified model converts the original sequence $s = s_1, s_2, \ldots, s_n$ into a corresponding sequence s' based on the hydrophobicity of the amino acids ($s_i^{'} \in \{H, P\}, \forall i \in \{1, 2, \ldots, n\}$).In the following example, we present a protein sequence of ten amino acids and its corresponding representation in the H-P model.

Primary structure $s =$ `Gln Asn Ala Ala Tyr Phe Gly Ala His Val`

The corresponding H–P sequence is:

$$s' = \texttt{P P H H P H P H P H}$$

Representation and Encoding of Solutions

Figure 1 illustrates the local neighborhood of a lattice position (x, y) on a two-dimensional square grid. In this representation, each point is surrounded by four direct neighbors located horizontally and vertically around it. These neighbors are given by

$$\{(x+1,\, y), (x,\, y+1), (x-1,\, y), (x,\, y-1)\},$$

corresponding to the Right, Up, Left, and Down directions, respectively.

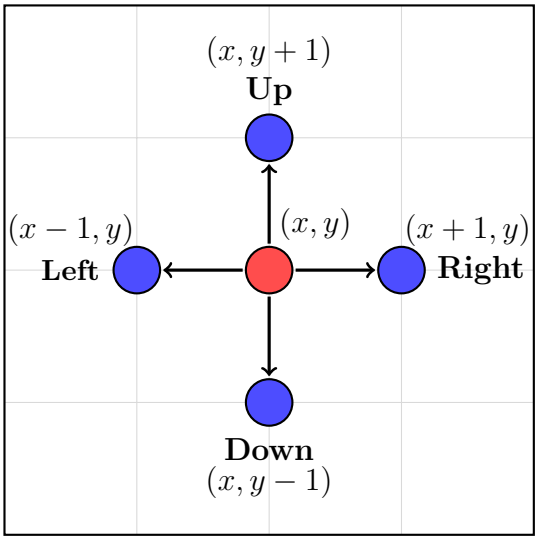

Figure 1: Illustration of the four neighbors of given node position (x, y) in 2D sequare lattice.

A feasible conformation of a protein consisting of n amino acids on a two-dimensional square lattice can be represented as a sequence of $n-1$ lattice moves forming a *self-avoiding walk*. That is, the folding path cannot revisit any lattice position previously occupied [4].

Let $\mathcal{P} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ be the set of lattice coordinates occupied by the n amino acids. A conformation is considered valid if and only if it satisfies the following constraints:

1. Each amino acid occupies exactly one lattice node, so we have:

$$\text{for all } i \neq j,\ (x_i, y_i) \neq (x_j, y_j).$$

2. No lattice node hosts more than one amino acid, ensuring that the walk is self-avoiding.

3. Consecutive amino acids in the sequence occupy adjacent lattice nodes, which imposes the geometric constraint :

$$|x_{i+1} - x_i| + |y_{i+1} - y_i| = 1, \quad i = 1, 2, \ldots, n-1.$$

Objective Function

The quality of a protein folding in the H-P model is quantified by the number of hydrophobichydrophobic (H-H) contacts in the conformation. To achieve an optimal folding, the number of such contacts should be maximized. Let s be a protein sequence of length n. The energy of a valid conformation $c(s)$ for s is defined by the following mathematical expression [4]:

$$E(c(s)) = -\sum_{i=1}^{j=n-1} \sum_{j=i+1}^{j=n} x_{i,j} y_{i,j},$$

where

$$x_{ij} = \begin{cases} 1, & \text{if } s_i = H \text{ and } s_j = H, \\ 0, & \text{otherwise.} \end{cases}$$

And,

$$y_{ij} = \begin{cases} 1, & \text{if amino acids } i \text{ and } j \text{ are in topological (adjacent) positions,} \\ 0, & \text{otherwise.} \end{cases}$$

Two amino acids are considered to be in *topological contact* if they occupy neighboring positions on the lattice, but are not consecutive along the protein sequence. In the H-P lattice model, the goal of protein folding prediction (PFP) is to find a valid conformation of a given protein sequence s that maximizes the number of H-H topological contacts, thus minimizing the total energy of the structure. As an example, the conformation shown in figure 2 corresponds to a feasible folding of the HP protein sequence given by:

$$s = \text{H P P P H H H H H H P H P P H P P H P H H H.}$$

This conformation contains 8 H-H topological contacts (i.e., energy of this conformation is $E(s) = -8$).

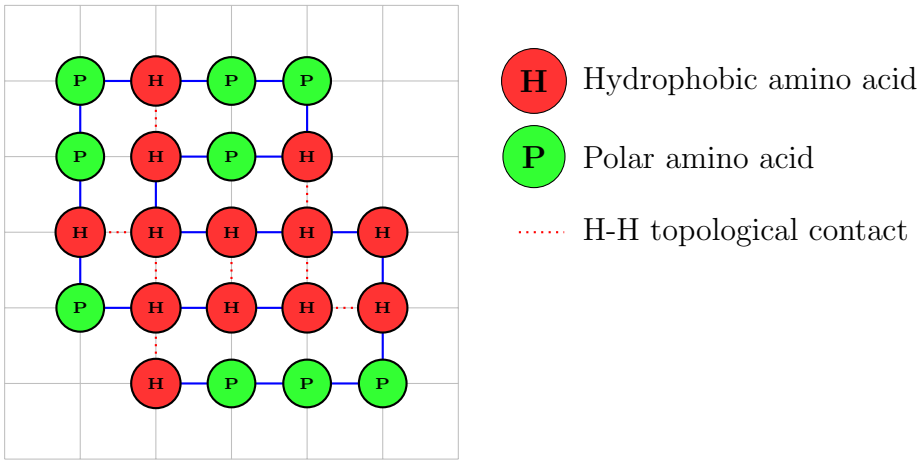

Figure 2: Representation of a folded HP protein sequence on a two-dimensional square lattice.

3.1 Mathematical Modeling of the PFP Problem on a 2D square Lattice

This section introduces a mathematical programming framework for the protein folding problem (PFP) within a two-dimensional sequare lattice. The proposed formulation is designed to be compatible with standard optimization modeling tools and solvable using existing mathematical programming solvers.

The 2D square lattice structure is characterized by the fact that each lattice node is adjacent to four neighboring positions, defined with respect to a chosen coordinate system. Let (i, j) denote a lattice coordinate. Given a protein sequence of length n, we define a binary decision variable y_{ij}^k to represent the spatial assignment of amino acids:

$$x_{ij}^k = \begin{cases} 1, & \text{if amino acid } k \text{ occupies position } (i, j), \\ 0, & \text{otherwise.} \end{cases}$$

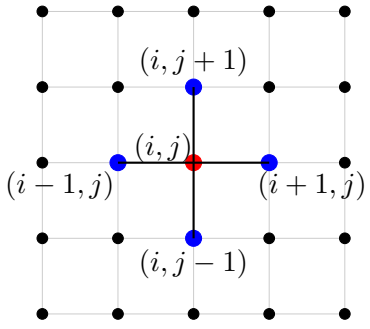

Figure 3: Illustration of the four-neighborhood of a node (i, j) in a 2D lattice.

3.1.1 Constraints

We consider a 2D square lattice where each node has four adjacent neighbors. The first amino acid of the sequence is fixed at the central position (n, n):

$$x_{nn}^1 = 1.$$

The feasibility of a solution is ensured through the following constraints:

- **Path length constraint:** A valid conformation must place all n amino acids on the lattice:
$$\sum_{k=1}^{n} \sum_{i=1}^{2n} \sum_{j=1}^{2n} x_{ij}^k = n.$$
- **Uniqueness constraint:** Each lattice position can host at most one amino acid:
$$\sum_{k=1}^{n} x_{ij}^k \leq 1, \quad \forall i, j \in \{1, \ldots, 2n\}.$$
- **Connectivity constraint (square lattice):** An amino acid at position $k+1$ must be placed on a node adjacent to the position of amino acid k. In the square lattice, each node has four neighbors:
$$x_{ij}^{k+1} \leq x_{i-1,j}^k + x_{i+1,j}^k + x_{i,j-1}^k + x_{i,j+1}^k, \quad \forall i, j \in \{1, \ldots, 2n\},\ \forall k \in \{1, \ldots, n-1\}.$$

3.1.2 Objective Function

Let θ_k be a binary parameter representing the type of amino acid:

$$\theta_k = \begin{cases} 1, & \text{if amino acid } k \text{ is hydrophobic (H)}, \\ 0, & \text{if amino acid } k \text{ is hydrophilic (P)}. \end{cases}$$

The objective is to maximize the number of non-consecutive hydrophobic contacts:

$$\max(Z) = \frac{1}{2} Z^* - \sum_{k=1}^{n-1} \theta_k \theta_{k+1},$$

where

$$Z^* = \max_y \left\{ \sum_{i=1}^{2n} \sum_{j=1}^{2n} \left(\sum_{k=1}^{n} \theta_k y_{ij}^k \right) \left(\sum_{k=1}^{n} \theta_k \left(x_{i-1,j}^k + x_{i+1,j}^k + x_{i,j-1}^k + x_{i,j+1}^k \right) \right) \right\}.$$

Remarks: This formulation ensures that the protein conformation remains within a bounded lattice defined over $[1, 2n] \times [1, 2n]$, with a fixed starting point at (n, n). The maximum spatial extension is limited by the sequence length, as the longest possible path requires at most $n - 1$ moves.

Although the model has a space complexity of $\mathcal{O}(n^2)$, it remains computationally tractable due to the binary nature of the decision variables.

4 A New Hybrid Genetic Algorithm for the PFP Problem

Here, We propose an efficient hybrid genetic algorithm, HGAETS (see Algorithm 1), to solve the protein folding problem (PFP) on a 2D square lattice. This algorithm combines the genetic algorithm (GA) with An Enhanced Tabu Search using Lévy Flight Perturbations (ETS). The motivation for this hybridization is that GA is highly effective at exploring the search space and identifying new regions, whereas ESA excels at exploiting local neighborhoods, moving from a current solution to its best neighbor. By combining these complementary strengths, HGAETS achieves a well-balanced trade-off between diversification and intensification, allowing the search space to be explored more efficiently. The algorithm begins by generating a population P of feasible chromosomes m (i.e., feasible solutions). Over successive generations, the quality of these solutions is improved through two phases: an exploration phase, driven by the GA, followed by an improvement phase executed by the proposed improved Tabu Search. In each generation, genetic operators are applied to create a new population. The solutions selected for reproduction undergo crossover, according to the crossover rate P_c, and are chosen using the tournament selection technique. A mutation operator is applied with probability $P_m = 1 - P_c$ to maintain diversity in the population. Each offspring produced in this phase is then used as the initial solution in the improved tabu search, applied with a fixed rate β (see Algorithm 2) to further enhance its quality. The population for the next generation is formed by selecting the best m chromosomes from the current population and the newly generated offspring. To introduce additional diversity, a rotation operator is applied to the new population before proceeding to the next generation. This cycle of exploration, improvement, and diversification is repeated until a stopping condition is met, ensuring that the algorithm efficiently navigates the search space while continuously improving the quality of the solution.

4.1 Chromosome Structure and Population Initialization

Given an HP protein sequence of length n, a feasible conformation on the 2D square lattice can be represented by a sequence move of $n-1$ movements in the lattice $(m_2, m_3, \ldots, m_n)$, where each movement $m_i \in \{R, L, U, D\}$ corresponds respectively to a step to the *Right*, *Left*, *Up*, or *Down* on the lattice. The first amino acid is placed at a fixed initial lattice position (x_1, y_1). For each amino acid i, its lattice coordinates are determined from the position of the preceding amino acid (x_{i-1}, y_{i-1}) and the movement assigned to it m_i. Formally, the position of a given amino acid i is given by

$$(x_i, y_i) = (x_{i-1}, y_{i-1}) + \Delta(m_i),$$

where the movement vector $\Delta(m_i)$ is defined as

$$\Delta(R) = (1, 0), \quad \Delta(L) = (-1, 0), \quad \Delta(U) = (0, 1), \quad \Delta(D) = (0, -1).$$

This representation allows the full conformation to be reconstructed from the initial position and the sequence of lattice movements. For the initial population, we generate m random solutions as follows. We begin by placing the first amino acid at a fixed initial

Algorithm 1 Suggested hybrid HGAETS algorithm

Input: Problem instance (i.e., HP protein sequence) I.

Output: The positions $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ of the n amino acids in the 2D sequare lattice.

Begin
Initialization:
$P \leftarrow$ Generate a initial population of m feasible Chromosomes.
Evaluate the Chromosomes using the fitness function (i.e., Energy function).
while the stop criterion is not checked **do**
Generate a new offspring population P_{new} by the flowing steps:
$k \leftarrow 0,\ P_{new} \leftarrow \{\emptyset\}$;
while $k \leq m$ **do**
$(s_1, s_2) \leftarrow$ Select two parents from P using Exponential Rank Selection technique.
if $rand_1 < p_c$ **then**
$(of_1, of_2) \leftarrow$ two solutions(offsprings) obtained by applying crossover operator to the selected parents.
else
$(of_1, of_2) \leftarrow$ two solutions(offsprings) obtained by applying mutation operator to the selected parents.
end if
if $rand_2 < \beta$ **then**
$(of_1, of_2) \leftarrow$ Improve the quality of the obtained offsprings (of_1, of_2) using the suggested ETS given in algorithm 2.
end if
$P_{new} \leftarrow P_{new} \cup \{(of_1, of_2)\}$.
$k \leftarrow k + 2$.
end while
end while
$P \leftarrow$ the m best solution of $P \cup P_{new}$.
end while
end while
end

position on the lattice. Then, for each amino acid i, a direction is randomly selected from the four possible movements. If the resulting position is already occupied by a previously placed amino acid (i.e., it creates a cycle), another direction is chosen. If all four directions lead to occupied positions, the partial conformation is discarded and a new solution is generated from the beginning.

4.2 Exponential Rank Selection

In our algorithm, parent chromosomes for the reproduction phase are selected using exponential rank selection (ERS). ERS improves upon linear rank selection by reducing its slow convergence. The main idea is to assign selection probabilities based on the relative ranking of chromosomes, giving higher-ranked chromosomes a greater chance of being chosen.

For minimization problems, such as minimizing energy in our problem, chromosomes are ranked from the lowest fitness (best) to the highest fitness (worst). The probability of selection of the chromosome at rank i is

$$p_i = \frac{(1-r)\, r^{i-1}}{1-r^m}, \quad i = 1, \ldots, m, \tag{1}$$

where m is the size of the population and $r \in (0, 1)$ controls the selection pressure. Larger values of r increase the likelihood of selecting top-ranked chromosomes while still allowing lower-ranked chromosomes to be chosen.

For example, consider a population of $m = 4$ chromosomes with fitness values: -32, -18, -45, and -16. Ranking from lowest to highest fitness gives:

Rank	Chromosome	Fitness
1	Chromosome 3	-45
2	Chromosome 1	-32
3	Chromosome 2	-18
4	Chromosome 4	-16

Using ERS with $r = 0.6$, the selection probabilities are:

$$\begin{aligned}
p_1 &= \frac{(1-0.6)\cdot 0.6^0}{1-0.6^4} = \frac{0.4}{0.8704} \approx 0.460,\\
p_2 &= \frac{0.4\cdot 0.6^1}{0.8704} = \frac{0.24}{0.8704} \approx 0.276,\\
p_3 &= \frac{0.4\cdot 0.6^2}{0.8704} = \frac{0.144}{0.8704} \approx 0.165,\\
p_4 &= \frac{0.4\cdot 0.6^3}{0.8704} = \frac{0.0864}{0.8704} \approx 0.099.
\end{aligned}$$

As expected, the lowest energy chromosome (-45) has the highest probability of selection, while higher energy chromosomes still retain a chance, preserving diversity in the population.

4.3 Crossover Operator

In our method, the 1-point crossover is applied to parent chromosomes that encode protein folding paths in a 2D lattice using four directional symbols: L (Left), R (Right), U (Up), and D (Down). This operator produces offspring by exchanging portions of the two parent sequences, allowing a combination of features from both and promoting exploration of new folding conformations. A random crossover point $c \in \{2, \ldots, n\}$ is selected along the sequence. The first segment of each parent before the crossover point remains unchanged, while the remaining segment is swapped between the parents. This results in two children, each inheriting parts of both directional information of both parents. The crossover is performed with a high probability p_c, promoting regular mixing of genetic material between parents.

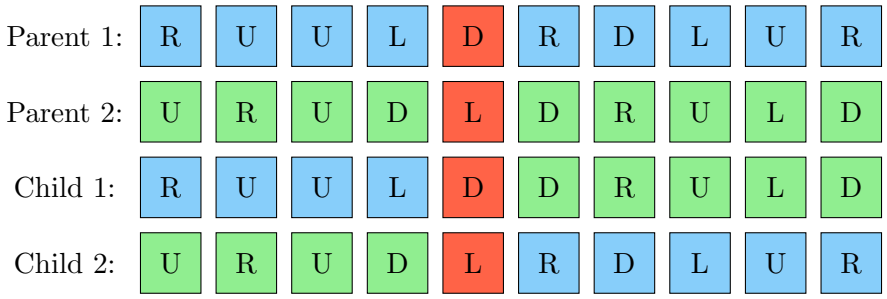

Figure 4: Illustrative example of 1-point crossover operator applied to two parent chromosomes with cut Point $c = 5$ marked ind red color.

4.4 Mutation Operator

The mutation operator introduces small random changes in a chromosome to explore new protein conformations while maintaining most of the inherited structure. In our method, a random position $p \in \{1, \ldots, n\}$ is selected in the sequence, and the direction at this position is replaced by a different one chosen randomly from the set $\{L, R, U, D\}$. By applying a mutation with a low probability $p_m = 1 - p_c$, the algorithm preserves most of the parent's structure, but adds minor changes to enhance the diversity of the population.

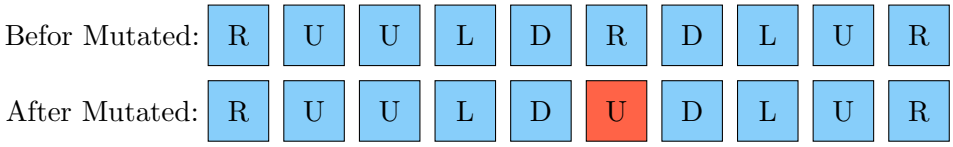

Figure 5: Example of mutation operator applied to a chromosome. The mutated position $p = 6$ is marked in red.

4.5 Improved Tabu Search Algorithm

To enhance the quality of solutions produced by the genetic algorithm, we integrate a Tabu Search (TS) procedure specifically designed for lattice-based protein structure

optimization. The method controls the selection of amino acids used for neighborhood exploration, while lattice-based diagonal moves combined with direction replacement are used exclusively to generate neighboring conformations.

Starting from a current solution s, the algorithm iteratively selects an amino acid as a pivot. The neighborhood set $\mathcal{N}(s)$ is defined as the set of candidate solutions generated by replacing the current direction of the selected amino acid with each of the three other possible lattice directions and applying the corresponding diagonal move to reposition the amino acid. All moves are performed under chain connectivity, self-avoidance, and lattice feasibility constraints to ensure valid conformations.

To prevent repeated selection of the same structural components, a tabu list is maintained as a First-In-First-Out (FIFO) queue of fixed length L. This list stores the indices of the last L selected amino acids. During each iteration, amino acids present in the tabu list are temporarily forbidden from selection, and no diagonal moves are applied to them. After executing a diagonal move, the selected amino acid is added to the end of the FIFO list, and the oldest entry is removed if the list exceeds its capacity. Among all candidate solutions generated from non-tabu amino acids, the algorithm selects the one with the minimum energy, even if it does not improve the current solution. By restricting the tabu memory to amino acid selection rather than specific moves or entire solutions, the proposed TS strategy achieves a balance between intensification and diversification. Intensification is achieved through systematic exploration of the local neighborhood via diagonal moves, while diversification is induced by temporarily forbidding recently used amino acids, promoting exploration of alternative conformational regions. Overall, this memory-guided Tabu Search procedure provides a deterministic, reproducible, and efficient local search mechanism that significantly improves the quality of offspring solutions within the evolutionary optimization framework.

4.5.1 Lévy-Flight-Based Discrete Operator

Lévy flights are a class of random walks characterized by a heavy-tailed step-length distribution, enabling a natural balance between frequent short moves and occasional long jumps. The step size s follows:

$$L(s) \sim s^{-\lambda}, \quad 1 < \lambda \leq 3. \tag{2}$$

While Lévy flights are commonly used in continuous optimization to update solutions as

$$X_{t+1} = X_t + \alpha \cdot \text{Lévy}(\lambda), \tag{3}$$

their direct application is not suitable for lattice-based protein structure prediction, which is inherently combinatorial. To address this, we propose a discrete Lévy-flight-based operator acting on direction sequences.

Each conformation is encoded as:

$$S = (d_1, d_2, \ldots, d_N), \quad d_i \in \{R, L, U, D\}.$$

Local transformations rely on predefined directional relationships (opposite, clockwise, and counterclockwise), allowing controlled modifications while preserving structural consistency (see Table 1)[20].

Algorithm 2 The Suggested Tabu Search Algorithm (ETS)

Input: Instance problem I and an initial solution s
Output: The best obtained solution s_{opt} for I

Begin
  Initialization:
  $E(s) \leftarrow$ Evaluate s using the energy function
  $s_{opt} \leftarrow s$
  $E(s_{opt}) \leftarrow E(s)$
  $TL \leftarrow \emptyset$
  $L \leftarrow$ Tabu tenure length
  $k \leftarrow 0$
  $g \leftarrow 0$
  While $g <$ maximum number of iterations (T_{max}) **do**
    $g \leftarrow g + 1$
    $i \leftarrow$ Generate a random amino acid index in $\{1, 2, \ldots, N\}$
      While the selected index $i \in TL$ **do**
        $i \leftarrow$ Generate another random amino acid index in $\{1, 2, \ldots, N\}$
      End While
    $N(s) \leftarrow$ Generate a neighborhood set for s applying the diagonal moves and direction replacement to the selected amino acid i
    Evaluate all $s' \in N(s)$
    $s' \leftarrow$ Select the best generated solution in $N(s)$
    $s \leftarrow s'$
    $E(s) \leftarrow E(s')$
    Update Tabu List TL:
      If $|TL| > L$ **then**
        Remove the oldest selected amino acid
      End If
    Insert i into TL
    If $E(s) < E(s_{opt})$ **then**
      $s_{opt} \leftarrow s$
      $E(s_{opt}) \leftarrow E(s)$
      $k \leftarrow 0$
    Else
      $k \leftarrow k + 1$
    End If
    If $k = k_{max}$ **then**
      $s \leftarrow$ Generate a diversified solution by applying Lévy flight to s
      $E(s) \leftarrow E(s)$
      $k \leftarrow 0$
    End If
  End While
Return s_{opt}
End Algorithm

Original Direction	Opposite	CW	CCW
Right (R)	L	D	U
Left (L)	R	U	D
Up (U)	D	R	L
Down (D)	U	L	R

Table 1: Directional transformation rules for amino acid folding in the 2D lattice (local search operators).

The proposed operator proceeds by sampling a Lévy step s, determining a subsequence length

$$L = \min(\lceil |s| \rceil, N - 1),$$

and selecting a starting index $i \in [1, N-L]$. The corresponding subsequence $(d_i, \ldots, d_{i+L-1})$ is then modified using either opposite or directional rotation (CW or CCW). This mechanism enables adaptive perturbations: small subsequences refine local structures, while larger ones introduce significant structural changes, enhancing exploration.

Illustrative Example Consider the sequence

$$S = (R, U, U, L, D, R, U, L, D).$$

For a sampled value $s = 4.6$, we obtain $L = 5$. Selecting $i = 3$ yields the subsequence (U, L, D, R, U), which is then transformed to generate neighboring solutions as summarized in Table 2.

Transformation	**The Obtained Solution**
Original	$(R, U, U, L, D, R, U, L, D)$
Opposite	$(R, U, U, R, D, L, U, L, D)$
Clockwise (CW)	$(R, U, R, U, L, D, R, L, D)$
Counterclockwise (CCW)	$(R, U, L, D, R, U, L, L, D)$

Table 2: Candidate solutions generated from the selected subsequence.

5 Experimental results and discussions

This section presents an experimental evaluation of the proposed HGA–ETS algorithm and compares its performance with several state-of-the-art methods applied to the Protein Folding Problem (PFP) in the two-dimensional square lattice model. The comparison is carried out using standard benchmark instances listed in Table 3. Such benchmark instances are commonly used in the literature to assess the effectiveness of algorithms designed for the protein folding problem. For each instance, the table provides the index, sequence length, and the corresponding H-P model representation. In the adopted representation, compact notation is used to describe repeated subsequences; for instance, the subsequence $HPHP$ is written as $(PH)^2$. Each experiment was repeated 20 times to ensure statistical reliability. All experiments were conducted on a computer equipped with an Intel Core i5 processor and 4 GB of RAM, with Python used as the programming

environment. Table 4 reports the parameter settings used for the suggested HGAETS algorithm.

Seq.	**Length(N)**	**Protein sequence (H-P model)**	E_{min}
1	20	$(HP)^2PH(HP)^2(PH)^2HP(PH)^2$	-9
2	24	$H^2P^2(HP^2)^6H^2$	-9
3	25	$P^2HP^2(H^2P^4)^3H^2$	-8
4	36	$P(P^2H^2)^2P^5H^5(H^2P^2)^2P^2H(HP^2)^2$	-14
5	48	$P^2H(P^2H^2)^2P^5H^{10}P^6(H^2P^2)^2HP^2H^5$	-23
6	50	$H^2(PH)^3PH^4PH(P^3H)^2P^4(HP^3)^2HPH^4(PH)^3PH^2$	-21
7	60	$P(PH^3)^2H^5P^3H^{10}PHP^3H^{12}P^4H^6PH^2PHP$	-36
8	64	$H^{12}(PH)^2((P^2H^2)^2P^2H)^3(PH)^2H^{11}$	-42

Table 3: Benchmark instances in the 2D square lattice model.

Parameter	**Value**
Population size m	100
The crossover rate p_c	0.8
The step-length in lévy Flights λ	$\frac{3}{2}$
The tabu list length L	$\frac{N}{3}$
The Tabu Search application rate β	0.3
Tne number of iterations in Tabu search T_{max}	10000
The maximum number of generations $G_{\max}$	1000

Table 4: Parameter settings used in the suggested HGAETS algorithm.

Table 5 summarizes the best energy values (i.e., minimal energy configurations) obtained by the proposed HGAETS, and compares them with several state-of-the-art methods for the H-P model in the 2D square lattice, including the Genetic Algorithm (PSPGA [21] and GA [31]), the improved genetic algorithm CCGA proposed in [29]. Ant Colony Optimization (ACO) [27], Monte Carlo (MC) methods [18], the protein folding framework called Folding Zero (Column Fzeros), based on deep reinforcement learning presented in [17], the deep reinforcement learning (DRL) algorithm [35], and the differential evolutionary algorithm (DE) presented in [14]. The comparative results presented in Table 5 demonstrate the effectiveness of the proposed HGA-ETS algorithm for solving the HP protein folding problem on the 2D square lattice model compared with several state-of-the-art approaches from the literature.

Seq.	N	GA	ACO	MC	ACGA	PSPGA	ED	FZEROS	DRL	HGAETS
1	20	**-9**	**-9**	-8	**-9**	NA	**-9**	**-9**	**-9**	**-9**
2	24	**-9**	**-9**	-8	**-9**	NA	**-9**	-8	**-9**	**-9**
3	25	**-8**	**-8**	-7	**-8**	NA	**-8**	-7	**-8**	**-8**
4	36	-12	**-14**	-12	**-14**	NA	**-14**	-13	**-14**	**-14**
5	48	-22	**-23**	-18	-22	NA	**-23**	-18	**-23**	**-23**
6	50	**-21**	**-21**	-19	**-21**	**-21**	**-21**	-18	**-21**	**-21**
7	60	-34	-34	-31	-35	-32	-35	-32	NA	**-36**
8	64	-31	-32	-31	-38	-37	-39	NA	NA	**-42**
9	85	NA	NA	NA	-48	-46	NA	-49	NA	**-50**
10	100	NA	NA	NA	NA	-42	NA	NA	NA	**-46**

Table 5: Comparison of the best energy values obtained by HGAETS with those reported in the literature for the used benchmark sequences. NA: not available data.

According to the benchmark reference values, the optimal energies for the first eight instances are equal to -9, -9, -8, -14, -23, -21, -36, and -42, respectively. The proposed HGA–ETS algorithm successfully reaches all these optimal values, achieving the global optimum for 8 out of the 10 tested benchmark instances.

For the first six benchmark sequences, with lengths ranging from $N = 20$ to $N = 50$, HGA-ETS obtains the same optimal energy values reported by the best existing approaches. More precisely, the proposed method matches the optimal results obtained by GA, ACO, ACGA, ED, and DRL for instances 1, 4, 5, and 6. In addition, HGA-ETS clearly outperforms MC for all these instances, since MC remains trapped at higher energy levels equal to -8, -8, -7, -12, -18, and -19, respectively. These observations confirm the stability and robustness of the proposed hybrid search strategy on small and medium-sized benchmark problems.

The superiority of HGA-ETS becomes more significant as the sequence length increases. For the benchmark instance with $N = 60$, the proposed algorithm reaches the optimal energy value of -36, while GA and ACO only achieve -34, MC reaches -31, PSPGA and FZEROS obtain -32, and ACGA and ED remain limited to -35. Hence, HGA-ETS improves the best previously reported value by one energy unit and significantly surpasses several classical approaches.

A more pronounced improvement is observed for the sequence of length $N = 64$. In this case, HGA-ETS attains the optimal energy value of -42, whereas the closest competing method, PSPGA, reaches only -39. Similarly, ACGA obtains -38, ED achieves -37, ACO reaches -32, and both GA and MC remain limited to -31. These results highlight the strong exploration capability of the proposed approach and its ability to escape local optima in highly complex search spaces.

For the larger benchmark instances with $N = 85$ and $N = 100$, the optimal energy values

are −53 and −48, respectively. In both cases, HGA-ETS produces near-optimal solutions but does not fully reach the global optimum.

For $N = 85$, HGA–ETS obtains −50, which is close to the optimum (−53) with a gap of 3 units, while still outperforming FZEROS (−49), ACGA (−48), and PSPGA (−46). For $N = 100$, the proposed method achieves −46, compared to the optimal value −48, with a gap of 2 units and a clear improvement over PSPGA (−42) and other competing approaches. HGA-ETS remains highly competitive, showing strong performance even when the optimal solution is not reached.

Overall, the obtained results indicate that HGA-ETS provides a highly competitive optimization framework for the HP protein folding problem. The integration of genetic operators with the ETS-based improvement phase allows the algorithm to maintain an effective balance between diversification and intensification, leading to high-quality conformations and superior convergence performance compared with existing methods from the literature.

From Table 5, it is evident that the proposed HGAISA algorithm consistently outperforms the compared methods in terms of the best energy values. In particular, the improved version of simulated annealing (SA) provides significantly better solutions than the standard version, especially for larger sequences, highlighting the efficiency and robustness of HGAISA in solving the protein folding problem. The superior performance of the proposed HGA-ETS algorithm is primarily due to the diversification strategy incorporated in its design, which helps the algorithm escape from local optima. While ISA is able to find the optimal solutions for instances 1 to 5, it fails to reach the optimal solution for instances 8 and 9. In contrast, HGAISA successfully finds the optimal solution for all instances, including the larger sequences.

Compared to ACO, MC, and GA algorithms, HGA-ETS demonstrates superior performance. Although all mentioned algorithms can predict optimal or near-optimal conformations for short to medium-length sequences, the proposed algorithm achieves a significant improvement in terms of best energy values for larger sequences, particularly for instances 8 and 9.

Table 6 presents the stability comparison of the implemented GA, ETS, and HGA-ETS algorithms against the benchmark method ED, evaluated over 50 independent runs for each instance. In this table, the numbers given in brackets represent the number of executions (out of 50 independent runs) in which the corresponding method obtained its best solution for each instance. For a statistically fair comparison, GA and HGA-ETS were executed using the same initial population and the same number of generations. This ensures that performance differences are due to the algorithmic design rather than initialization bias or computational effort. Overall, the results show that the proposed methods significantly improve stability compared to GA and ED. The HGA-ETS (Hybrid Genetic Algorithm–ETS), which combines GA and ETS mechanisms, achieves the best overall performance in terms of robustness. It consistently obtains very high success rates (close to 50/50 in most small and medium instances) and maintains almost identical best and mean values for several cases (1–4 and 6). Even for larger instances (7–8), HGA-ETS preserves strong stability, with relatively small deviations between best and mean values,

confirming the benefit of hybridizing GA exploration with ETS intensification.

The ETS method also demonstrates strong and consistent behavior. It clearly outperforms GA and ED in most instances, showing improved success rates and reduced variability. However, its stability remains slightly lower than that of HGA-ETS, particularly on larger instances where the gap between best and mean values becomes more noticeable. In contrast, ED shows competitive performance on small instances but its stability decreases as the problem size increases, with lower success rates and larger variability compared to the proposed methods. Finally, GA exhibits the weakest performance overall, with the lowest stability and highest sensitivity to random initialization. The results confirm that hybridizing GA with ETS in the proposed HGA-ETS framework significantly enhances stability, while ETS alone already provides a substantial improvement over classical GA and ED.

	ED		**GA**		**ETS**		**HGA-ETS**	
Seq.	Best	Mean	Best	Mean	Best	Mean	Best	Mean
1	**-9(50)**	**-9**	-9(29)	-8.58	**-9(50)**	**-9**	**-9(50)**	**-9**
2	**-9(50)**	**-9**	-9(37)	-8.74	**-9(50)**	**-9**	**-9(50)**	**-9**
3	**-8(50)**	**-8**	-8(30)	-7.6	**-8(50)**	**-8**	**-8(50)**	**-8**
4	**-14(50)**	**-14**	-12(17)	-11.14	-14(33)	-13.32	**-14(50)**	**-14**
5	-23(45)	-22.88	-21(11)	-15.46	-23(15)	-21.10	**-23(47)**	**-22.94**
6	**-21(50)**	**-21**	-19(13)	-15.87	-21(11)	-18.64	**-21(50)**	**-21**
7	-35(42)	-34.82	-33(9)	-27.11	-34(17)	-32.36	**-36(44)**	**-34.33**
8	-39(40)	-38.80	-32(6)	-24.33	-37(14)	-33.47	**-42(43)**	**-41.72**

Table 6: Comparison of stability (best and mean results) of HGA-ETS against GA, ED and the ETS.

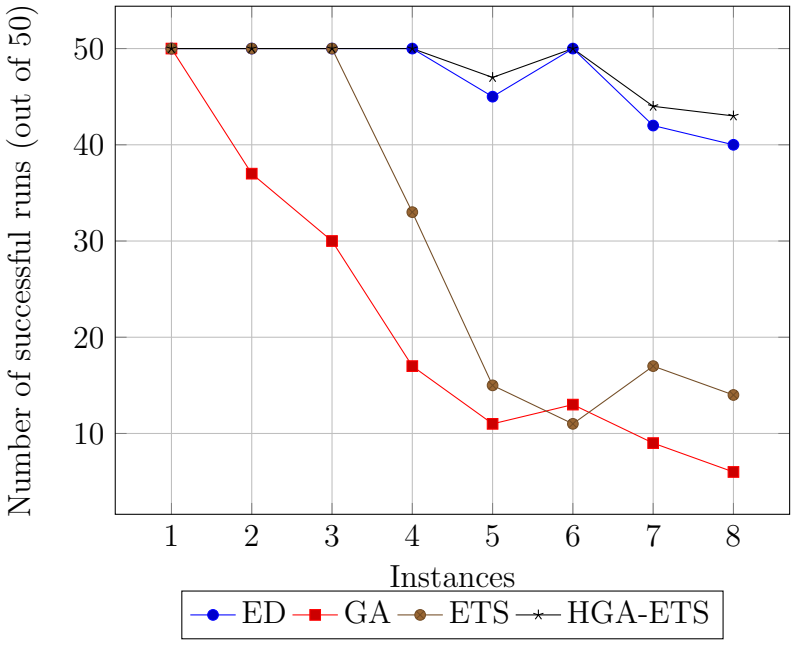

Figure 6: Stability comparison based on number of successful runs (out of 50).

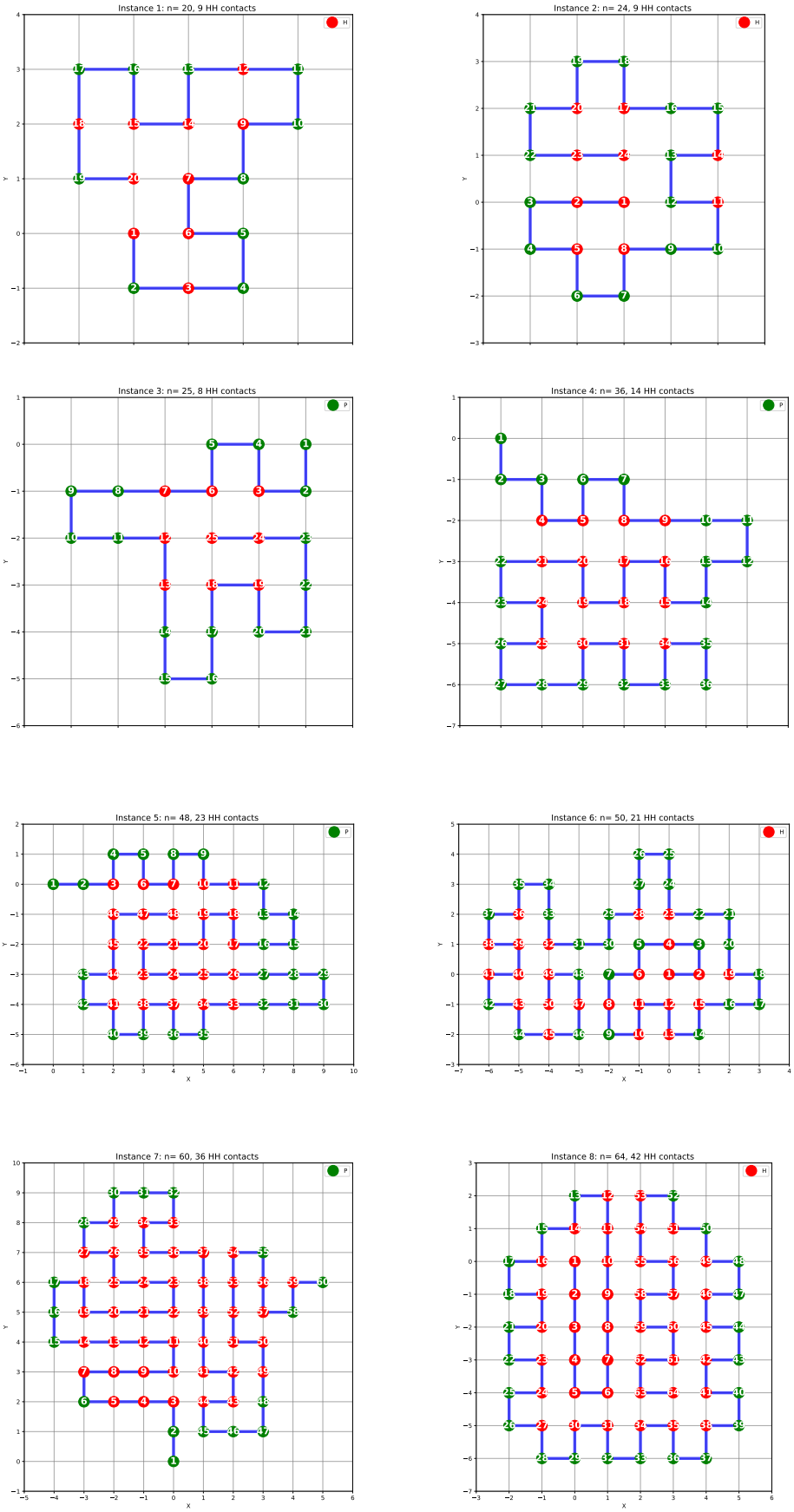

Figure 7: The best solutions obtained by the proposed HGA-ETS on the first eight benchmark instances.

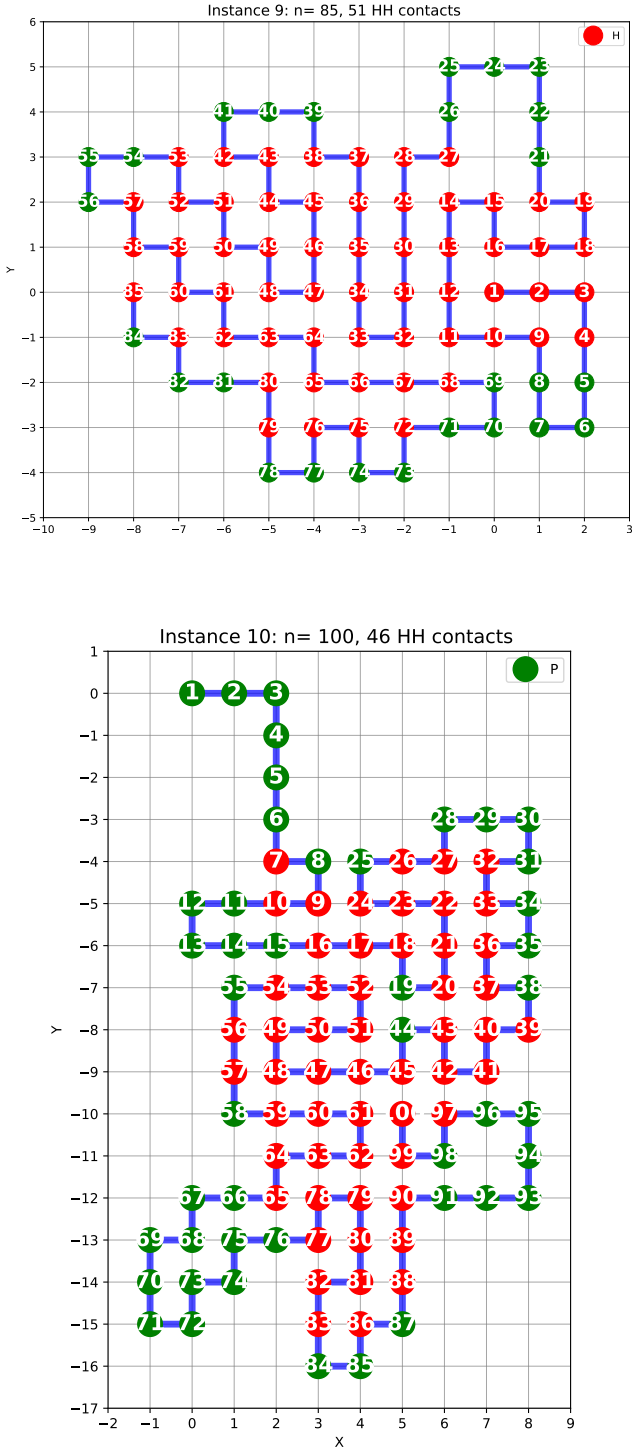

Figure 8: The best solutions obtained by the proposed HGA-ETS on the last two benchmark instances.

6 Conclusion

Protein structure prediction represents one of the most challenging optimization problems due to its high computational complexity. Developing efficient approaches capable of generating an optimal or near-optimal three-dimensional structure from a protein's primary sequence can significantly contribute to a better understanding of biological processes and facilitate the treatment of numerous diseases. In this paper, we proposed an efficient hybrid algorithm to address the protein structure prediction problem under the simplified HP model. The proposed approach combines a Genetic Algorithm (GA) with an Enhanced Tabu Search (ETS) procedure, thereby benefiting from the complementary strengths of global exploration and local exploitation. The experimental results demonstrate that the proposed HGA-ETS algorithm is capable of consistently producing high-quality solutions and outperforming most well-established state-of-the-art methods in terms of both solution quality and statistical robustness. These promising results can be attributed to the effective balance achieved between diversification and intensification throughout the search process. Future research will focus on extending the proposed HGA-ETS algorithm to more realistic protein folding models that better reflect the characteristics of real-world biological systems. In addition, the proposed framework may be adapted to address other challenging combinatorial optimization problems. Finally, a parallel implementation of the algorithm will be investigated to further improve its scalability and computational efficiency when solving large-scale instances.

References

[1] Eric Alm and David Baker. Prediction of protein-folding mechanisms from free-energy landscapes derived from native structures. *Proceedings of the National Academy of Sciences*, 96(20):11305–11310, 1999.

[2] Christian B Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[3] Andrei Băutu and Henri Luchian. Protein structure prediction in lattice models with particle swarm optimization. In *International Conference on Swarm Intelligence*, pages 512–519. Springer, 2010.

[4] Nabil Boumedine and Sadek Bouroubi. Protein folding in 3d lattice hp model using a combining cuckoo search with the hill-climbing algorithms. *Applied Soft Computing*, 119:108564, 2022.

[5] Zeinab Breijyeh and Rafik Karaman. Comprehensive review on alzheimerâs disease: causes and treatment. *Molecules*, 25(24):5789, 2020.

[6] Jean Carruthers and Alastair Carruthers. Mad cows, prions, and wrinkles. *Archives of dermatology*, 138(5):667–670, 2002.

[7] Vincenzo Cutello, Giuseppe Morelli, Giuseppe Nicosia, and Mario Pavone. Immune algorithms with aging operators for the string folding problem and the protein folding

problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 80–90. Springer, 2005.

[8] Thomas Dandekar and Patrick Argos. Folding the main chain of small proteins with the genetic algorithm. *Journal of Molecular Biology*, 236(3):844–861, 1994.

[9] Charles J Epstein, Robert F Goldberger, and Christian B Anfinsen. The genetic control of tertiary protein structure: studies with model systems. In *Cold Spring Harbor symposia on quantitative biology*, volume 28, pages 439–449. Cold Spring Harbor Laboratory Press, 1963.

[10] Paul J Hagerman and Ignacio Tinoco Jr. From sequence to structure to function. *Current opinion in structural biology*, 6(3):277–280, 1996.

[11] WE Hart, N Krasnogor, DA Pelta, and J Smith. Protein structure prediction with evolutionary algorithms. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States); Sandia â¦, 1999.

[12] William E Hart and Sorin Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eights of optimal. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 157–168, 1995.

[13] Andrea Ilari and Carmelinda Savino. Protein structure determination by x-ray crystallography. *Bioinformatics: Data, Sequence Analysis and Evolution*, pages 63–87, 2008.

[14] Nanda Dulal Jana and Jaya Sil. Protein structure prediction in 2d hp lattice model using differential evolutionary algorithm. In *Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012*, pages 281–290. Springer, 2012.

[15] Rainer König and Thomas Dandekar. Improving genetic algorithms for protein folding simulations by systematic crossover. *BioSystems*, 50(1):17–25, 1999.

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

[17] Yanjun Li, Hengtong Kang, Ketian Ye, Shuyu Yin, and Xiaolin Li. Foldingzero: Protein folding from scratch in hydrophobic-polar model. *arXiv preprint arXiv:1812.00967*, 2018.

[18] Faming Liang and Wing Hung Wong. Evolutionary monte carlo for protein folding simulations. *The Journal of Chemical Physics*, 115(7):3374–3380, 2001.

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

[20] Cheng-Jian Lin and Shih-Chieh Su. Using an efficient artificial bee colony algorithm for protein structure prediction on lattice models. *International journal of innovative computing, information and control*, 8(3):2049–2064, 2012.

[21] Arash Mazidi and Fahimeh Roshanfar. Pspga: A new method for protein structure prediction based on genetic algorithm. *Journal of Applied Dynamic Systems and Control*, 3(1):9–16, 2020.

[22] Sergio Moreno-Hernández and Michael Levitt. Comparative modeling and protein-like features of hydrophobic–polar models on a two-dimensional lattice. *Proteins: Structure, Function, and Bioinformatics*, 80(6):1683–1693, 2012.

[23] David G Munoz and Howard Feldman. Causes of alzheimer's disease. *Cmaj*, 162(1):65–72, 2000.

[24] Alantha Newman. A new algorithm for protein folding in the hp model. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete Algorithms*, pages 876–884, 2002.

[25] Stanley B Prusiner. Detecting mad cow disease. *Scientific American*, 291(1):86–93, 2004.

[26] Bouroubi Sadek and Nabil Boumedine. A new hybrid genetic algorithm for protein structure prediction on the 2dtriangular lattice. *Turkish Journal of Electrical Engineering and Computer Sciences*, 29(2):499–513, 2021.

[27] Alena Shmygelska, Rosalia Aguirre-Hernandez, and Holger H Hoos. An ant colony optimization algorithm for the 2d hp protein folding problem. In *International workshop on ant algorithms*, pages 40–52. Springer, 2002.

[28] Alena Shmygelska and Holger H Hoos. An improved ant colony optimisation algorithm for the 2d hp protein folding problem. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 400–417. Springer, 2003.

[29] Ioan Sima, Daniela-Maria Cristea, Laszlo Barna Iantovics, and Virginia Niculescu. Acga a novel biomimetic hybrid optimisation algorithm based on a hp protein visualizer: An interpretable web-based tool for 3d protein folding based on the hydrophobic-polar model. *Biomimetics*, 10(11):763, 2025.

[30] Ron Unger and John Moult. Finding the lowest free energy conformation of a protein is an np-hard problem: proof and implications. *Bulletin of mathematical biology*, 55(6):1183–1198, 1993.

[31] Ron Unger and John Moult. Genetic algorithms for protein folding simulations. *Journal of molecular biology*, 231(1):75–81, 1993.

[32] Julie S Valastyan and Susan Lindquist. Mechanisms of protein-folding diseases at a glance. *Disease models & mechanisms*, 7(1):9–14, 2014.

[33] Bertus VAN DAN BURG, Bauke W Dijkstra, Gert Vriend, Bernard VAN DAR VINNE, Gerard Venema, and Vincent GH Eijsink. Protein stabilization by hydrophobic interactions at the surface. *European journal of biochemistry*, 220(3):981–985, 1994.

[34] KURT VVÜTHRICH. Protein structure determination in solution by nuclear magnetic resonance spectroscopy. *Science*, 243:4887, 1989.

[35] Kaiyuan Yang, Houjing Huang, Olafs Vandans, Adithya Murali, Fujia Tian, Roland HC Yap, and Liang Dai. Applying deep reinforcement learning to the hp model for protein structure prediction. *Physica A: Statistical Mechanics and its Applications*, 609:128395, 2023.