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

Availaible on line at http://www.liforce.usthb.dz
bulletin-liforce@usthb.dz
Bulletin du Laboratoire 04 (2019) 78 - 100

A new hybrid genetic algorithm for protein structure prediction on the 2D triangular lattice

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

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

nabil.doukou@gmail.com[1], sbouroubi@usthb.dz[2]

Abstract: The flawless functioning of a protein is essentially linked to its own three-dimensional structure. Therefore, the prediction of a protein structure from its amino acid sequence is a fundamental problem in many fields that draws researchers attention. This problem can be formulated as a combinatorial optimization problem based on simplified lattice models such as the hydrophobic-polar model. In this paper, we propose a new hybrid algorithm combining three different well-known heuristic algorithms: genetic algorithm, tabu search strategy and local search algorithm in order to solve the PSP problem. Regarding the assessment of suggested algorithm, an experimental study is included, where we considered the quality of the produced solution as the main quality criterion. Furthermore, we compared the suggested algorithm with state-of-the-art algorithms using a selection of well-studied benchmark instances.

Keywords: Protein Structure Prediction, 2D triangular Lattice, HP Model, Genetic Algorithm, Local Search, Tabu Search Strategy, Minimal Energy Conformation.

1 Introduction

In Molecular biology, the three dimensional structure of proteins is the most crucial indicator that determines their biological activity. The prediction of the tertiary structure of proteins, also called a conformation, is a fundamental problem in the fields of biology, physics, etc. This problem is famous for protein structure prediction and denoted by PSP. A small modification in the original conformation of any given protein or an error during its folding causes many serious diseases, such as Alzheimer and cow mad. The ideal solution to treat these diseases is to predict the tertiaries structures of these proteins from their primary structure [9]. The PSP is one of the hardest problems in computational biology, molecular biology, biochemistry and physics. Furthermore, The most efficient algorithm for feasible solution determination runs in an an considerable time required; whereas the correct functioning of a protein depends essentially on its minimal energy conformation. Among the existing models in literature, the most studied in PSP problem is the H-P model (Hydrophobic Polar, H-P) [16].In this model the free energy of a conformation is inversely proportional to the number of hydrophobic non-local bonds of H-H type (topological contacts H-H). This type of bonds occurs if two non-consecutive hydrophobic monomers occupy adjacent grid points in the lattice. Besides, each occurrence of this bond type reduces the value of global energy with one unit [17]. The PSP is an optimization problem where the aim is to find a confirmation c^* of a given protein sequence that minimizes the overall induced energy in all possible set of conformations C, i.e., a conformation c^* such that $E(c^*) = min\{E(c)/c \in C\}$[22], where $E(c)$ represent the energy function explained later in Section 2.2.

As we mentioned above, the H-H bonds reduces the induced energy. Hence, finding a minimal energy conformation (i.e., optimal conformation) amounts to find a conformation that maximizes the number of H-H contacts [17]. As one may expect, the resolution of this problem is quite difficult due to the exponential exploration of the NP-hard problem solving when the chain size is large enough. The problem is proven to be NP-hard even when restricted to its two dimensional representation [2].Hence, it is clearly impossible to enumerate all the possible conformations when the chain of amino acids is considerably large. Several metaheuristics were proposed to solve the PSP problem. In the two-dimensional square lattice, the first genetic algorithm has been introduced by Unger and Moult [25], and than followed by other versions, see [8, 13, 15]. Similarly, an ant colony optimization (ACO) algorithm has been used by Shmygelska et al. [20, 21, 22]. Moreover, the use of memetic algorithms was proposed in [14, 18]. Particle Swarm Optimization algorithms has been applied in [5]. Jiang, T. et al. proposed a hybrid approach combining tabu search and genetic algorithm [12]. The Immune algorithms are introduced by Cutello et al. in [6, 7]. A Clustered memetic algorithm with local heuristics have been introduced by Islam et al. in [11].

This paper is organized as follows: in the next section, we present the two dimensional triangular lattice using the H-P model, that we are interested in, where we present a corresponding 0-1 mathematical program while focusing on: the objective function and the encoding solutions. In Section 3 we present a selection of pertinent algorithms designed to solve the PSP problem in 2D triangular lattice model. In Section 3, we present a detailed description of the suggested hybrid algorithm. In Section 4, we present the exper-

imental study and the obtained results, where we compared our approach some existing approaches. Finally, we give, in Section 6, our main conclusions of this study.

2 Hydrophobic-polar model in a 2D triangular lattice Simplified

In H-P model, the twenty amino acids are represented by a mean of two letters H and P, in reference to their hydrophobicity chosen among the tow following options : hydrophobic or hydrophilic [16]. For any given sequence of n amino acids, the H-P model consists to transform this sequence into a chain $s = (s_1, s_2, \ldots, s_n)$ such that each element of the sequence $s_i \in \{H, P\}$, $i = \overline{1, n}$ represents the hydrophobicity of the corresponding amino acids in protein sequence:

$$s_i = \begin{cases} H & \text{if the amino acid } i \text{ is of hydrophobic type,} \\ P & \text{if the amino acid } i \text{ is of polar type.} \end{cases}$$

As it is shown in Figure 1, each vertex of the two dimensional triangular lattice has six neighbors. Hence, each monomer different from the first and the last element of the chain, i.e., monomers of rank $i = \overline{2, n-1}$, occupying any given position in the lattice can be at most in four topological contacts, in other words, it can form at most four bonds with its neighbors. Whereas maximal number of possible contacts that can occur in a monomer located either in the first or the last position is five [1]. The symbols $1, 2, 3, 4, 5, 6$ are used to encode the following movement directions on the two dimensional triangular lattice: Right-Up, Up, Left-Up, Left-Down, Down and Right-Down, respectively.

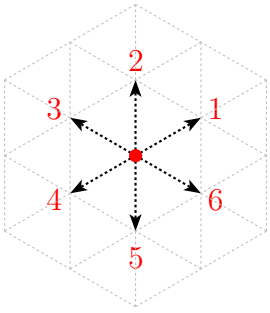

Figure 1: Illustration of the six neighbors of a node in the 2D triangular lattice model.

Figure 2 represents an feasible conformation in the 2D triangular lattice model for a protein sequence of 20 amino acids given in the HP model by $(HP)^2PH(HP)^2(PH)^2HP(PH)^2$. The green points represent the hydrophilic amino acids while the hydrophobic amino acids are represented in red. The energy of the conformation given in Figure 2 is $E(s) = -15$ (15 contacts of H-H type).

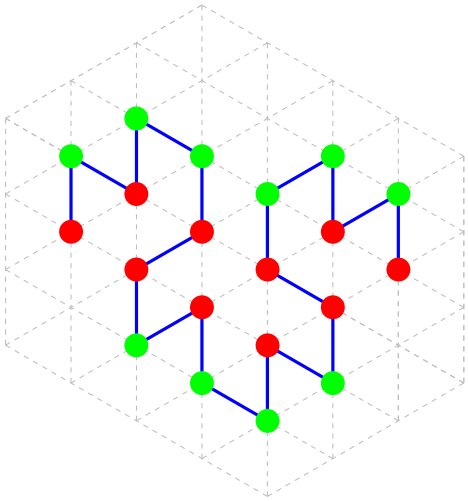

Figure 2: Representation of an feasible conformation for the sequence $(HP)^2PH(HP)^2(PH)^2HP(PH)^2$ in the 2D triangular lattice.

2.1 Encoding Solution

A feasible solution can be represented by the sequence of $n-1$ movements in the lattice called self-avoiding paths, (see Figure 1) that generate this conformation. For example, the corresponding movements vector to the solution S, given in Figure 2 is as follow:

$$mv(S) = [2, 6, 2, 6, 5, 4, 5, 1, 5, 6, 2, 6, 2, 3, 2, 1, 5, 1, 5].$$

The sequence $mv(S)$ allows us to deduce the position of each amino acid in the lattice.

2.2 Calculation of free energy

Given a feasible protein conformation of n amino acids and let s be its corresponding sequence in the HP model. The folding quality of this of this conformation is measured by the following energy function [17]:

$$E(s) = -\sum_{i=1}^{n-2}\sum_{j=i+2}^{n} \delta_{ij}\ x_{ij},$$

where δ_{ij} represents an indicator that determines whether both elements s_i and s_j are hydrophobic.

$$\delta_{ij} = \begin{cases} 1; & \text{if } (s_i = H) \wedge (s_j = H), \\ 0; & \text{otherwise,} \end{cases}$$

and

$$x_{ij} = \begin{cases} 1; & \text{if the amino acids } i \text{ and } j \text{ form an H-H topological contact,} \\ 0; & \text{otherwise.} \end{cases}$$

2.3 Mathematical program for the PSP

In the rest of this section, we present a mathematical program for the Protein Folding Problem (PSP) in its 2D triangular lattice model form. The aim here is to construct a linear mathematical program that can be implemented in modeling languages and compatible with the existing software solvers. Each node in a 2D triangular lattice blue located in a given position (i, j) has six neighbors represented in a grid on a canonical basis. Considering the following possible neighboring directions of (i, j) node:

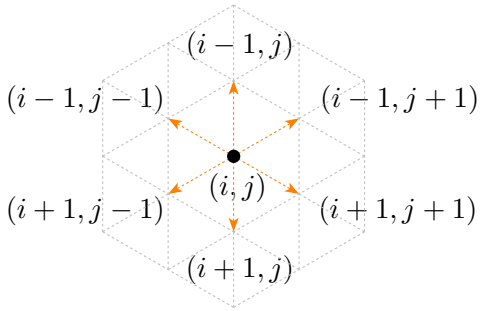

Figure 3: The six positions close to the position (i, j) in the 2D triangular lattice.

Let n be the length of the protein sequence, and let y_{ij}^k be a three dimensional variable such that:

$$y_{ij}^k = \begin{cases} 1, & \text{if the position } (i, j) \text{ contains the } k^{th} \text{ amino acid in the protien sequence,} \\ 0, & \text{else.} \end{cases}$$

2.3.1 Constraints

First, we fix the first amino acid in the protein sequence at position (n, n) as a starting point, i.e.,

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

Regarding the problem constraints, we can identify three different constraints that guarantees the admissibility of the resulting solution :

1- A path in the grid is a solution if it occupies exactly n nodes in the grid. This constraint can be written as follows:

$$\sum_{k=1}^{n}\sum_{i=1}^{2n}\sum_{j=1}^{2n} y_{ij}^{k} = n.$$

2- A node in the grid can contain at most one amino acid at the k^{th} position, hence:

$$\sum_{k=1}^{n} y_{ij}^{k} \leq 1,\ \forall i \in \{1,\ldots,2n\},\ \forall j \in \{1,\ldots,2n\}.$$

3- A node in the grid can contain the amino acid in position $k+1$, if and only if at least one of its neighboring nodes contains the k^{th} amino acid in the protein sequence:

$$y_{ij}^{k+1} \leq y_{i-1j+1}^{k}+y_{i-1j}^{k}+y_{i-1j-1}^{k}+y_{i+1j-1}^{k}+y_{i+1j}^{k}+y_{i+1j+1}^{k}, \forall i,j \in \{1,\ldots,2n\},\ \forall k \in \{1,\ldots,n-1\}.$$

2.3.2 The objective function

Let α_k be a numerical interpretation of any given amino acid into a binary value (i.e., representation in $\{0,1\}$), where:

$$\alpha_k = \begin{cases} 1 & \text{if the } k^{th} \text{ amino acid in the protein sequence is hydrophobic, i.e., H,} \\ 0 & \text{if the } k^{th} \text{ amino acid in the protein sequence is hydrophilic , i.e., P.} \end{cases}$$

Thus, the objective function can be calculated as follows:

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

where

$$\mathcal{Z}^{*} = \max_{y}\left\{\sum_{i=1}^{2n}\sum_{j=1}^{2n}\left(\sum_{k=1}^{n}\alpha_k y_{ij}^{k}\right)\left(\sum_{k=1}^{n}\alpha_k\left(y_{i-1j+1}^{k}+y_{i-1j}^{k}+y_{i-1j-1}^{k}+y_{i+1j-1}^{k}+y_{i+1j}^{k}+y_{i+1j+1}^{k}\right)\right)\right\}.$$

This mathematical model guaranties optimal solution which is included in the grid enclosed by the points $\{(1,1),(1,2n),(2n,1),(2n,2n)\}$, with a starting point (n,n). The choice of these limit points is based on the fact that in a path graph P_n on the grid, the maximal distance between the fixed starting node and the rest of the nodes (i.e., basically for the end node) is at most equals to $n-1$ movements. More precisely it represents the radius of the graph, which is the number of edges in the case of a path graph P_n. With the view to its spatial complexity $O(n^3)$, this model clearly has a rather high cost in terms of memory. However, it provides a rather good equilibrium with the computational complexity; since all variables are binary strings/arrays.

3 Existing 2D H-P protein prediction algorithms

Recently a number of metaheuristcs have been used to solve the PSP in the 2D triangular lattice model. In [10], the authors suggested a new hybrid algorithm, called Hybrid Genetic Algorithm (HGA). This latter enhances the performance of a classical GA implementation by reducing the encountered through the generational process. More specifically, it is clear that when the number of generations increases, the current conformations become very compact. Hence, the application of genetic operators produces invalid conformations with considerable number of collisions. The proposed approach consists in using the generalized short pull moves strategy to produce only valid conformations to avoid blocking GA search processes. The authors have shown considerable quality improvements when compared to a simple genetic algorithm implementation SGA [10]. Later on in [4], the authors proposed new approach based on the tabu search algorithm using a generalized local move (i.e., pull move) as to improve the landscape exploration and the quality of the produced solutions.. Also, two approaches are proposed in [17], including the Elite-based Reproduction Strategy-Genetic Algorithm (ERS-GA) ,and a Hybrid of Hill climbing and Genetic Algorithm called HHGA that combine the ERS-GA with a hill climbing algorithm.A new approach called IMOG has been proposed in [26], that combines ions motion optimization algorithm (IMO) with a Greedy algorithm (G), the obtained results shown that the IMOG algorithm has a good search ability and stability for the PSP problem using benchmark data sets.

4 Novel hybrid approach for PSP problem

The most commonly used hybridization in literature consists of combining two metaheuristics, one based on a single solution, known as s-metaheuristics, and the other based on a population, known as p-metaheuristics. The s-metaheuristics have proved their effectiveness for intensification, while the p-metaheuristics known by their exploration capacity. Thus, this type of hybridization allows to establish a balance between the diversification and the intensification of the search process [3, 19, 24].

In this work, we proposed a hybrid approach to solve the PSP problem, called GALSTS, in reference to the adopted combination of the following metaheuristics: genetic algorithm, local search, tabu search strategy. As all population-based approaches the first step of GALSTS is to generate an initial population P of m feasible solutions, in which the crossover operator is guided by a tabu list that allows to prohibiting previous movements in order to explore new regions in the search space. Each two selected parent produces a set of offsprings (i.e., f solutions) by applying the crossover procedure controlled by a tabu list that contains all k crossover points that are already used. This has allowed us to avoid cyclical movements.

In order to improve the quality of the children produced by the crossover process, each one of them is introduced with a probability p_m as an initial solution of a local search

algorithm. Such that the transition from a solution s to one of its neighbors s' is carried out by a random choice of an amino acid i and replacing its direction by another one among the five other possible directions. If the quality of s is better than s', then it becomes the courant solution for the next iteration. This process is repeated until the satisfaction of the stops criterion. This improvement phase has allowed us to use the information provided by parents more efficiently to produce high-quality solutions. The two solutions with the highest fitness value are introduced into the new population P_{new} if they not exist in this population previously, although are of lower quality than their parents to encourage exploration of the research space. The best m solutions of $P \cup P_{new}$ are replaced the individuals in the population P for the next generation. This approach is intended to avoid the rapid convergence towards local optima, and the wide diversification between the solutions and thus ensures the quality of the solutions during all the stages of the research space explorations, see Algorithm 1.

Algorithm 1 Suggested hybrid algorithm GALSTS

Require: A protein sequence of amino acids of size n.

Ensure: The best confirmation for the protein sequence.

begin

Initialization: $P \leftarrow$ The initial population of m solutions;

while the stop criterion is not checked **do**

Create a new population by the flowing steps:

$k \leftarrow 0,\ P_{new} \leftarrow \emptyset$;

while $k \leq m$ **do**

Select two parents (p_1, p_2) from P;

$offspring \leftarrow$ two solutions obtained by applying the algorithms (TS-LS) to the selected parents;

if $offspring$ does not exist in the new population P_{new} **then**

$P_{new} \leftarrow P_{new} \cup \{offspring\}$;

$k \leftarrow k + 1$;

end if

end while

$P \leftarrow$ the m best solution among $P \cup P_{new}$;

end while

end

4.1 The Initial Population

We start with an initial population of m randomly generated individuals. An initialization algorithm was proposed that allows to generate only valid conformations for the initial population of GALSTS, we use a list T that contains all the positions used for courant solution, we put the first amino acid at one point in the lattice and save their position in T. Then, for each amino acid we choose a random direction among the six directions as shown in Figure 1. If the selected direction generates an already occupied position (i.e., existing in T); we generate another direction different from the selected one. If all the six directions create an existing positions in T (i.e., all direction create invalid solution), we generate a new solution, see Algorithm 2.

4.2 Crossover Operator

The crossover procedure consists to combine two or more solutions, called parents, as to create other solutions, called offsprings. There are several types of the crossover operator, here we opted for a random 1-point, which consists to swap after selecting two parents p_1 and p_2, and generating one random point c_1, $1 < c_1 < n$, the parent subsequences limited by c_1 and n. As shown in Figure 4, the new two conformations (offsprings) f_1 and f_2, are obtained by combining p_1 and p_2 after generating a random cut point (here $c_1 = 5$). The energy value of offspring f_1 is $E(f_1) = -7$, lower than the energy values of its parents, $E(p_1) = -2$ and $E(p_2) = -5$.

4.3 Mutation Operator

Generally, it consists to modify some components, called genes, of an existing solution, in order to introduce more diversity into the solutions, it generally applied with low probability. In the proposed local search algorithm (see Algorithm 3), the neighbors of a given solution are defined in a similar way with the mutation operator, but the performed movement is accepted if it improves the quality of the courant solution. As shown in Figure 5, the new conformation s_m is obtained by exciting the current solution s at the mutation point 10, changing the direction from 3 to 6. The energy value of s_m is $E(s_m) = -7$, lower than the energy of s, $E(s) = -5$.

Algorithm 2 Generation algorithm for the initial population

Require: n, the number of amino acids in the protein sequence.

Ensure: A feasible confirmation for the protein sequence.

begin

Initialization: $T \leftarrow$ Table of position for each amino acids in lattice;

$T[1] \leftarrow (x_1, y_1)$, i.e., put the first monomer in one vertex of the lattice

$i \leftarrow 2$;

while $(i \leq n)$ **do**

$K \leftarrow \{1, 2, 3, 4, 6\}$; the set of all possible directions on the 2D triangular lattice

$t \leftarrow true$;

while $(t = true)$ **do**

$u \leftarrow$ Random direction generated from the list K;

$K \leftarrow K \backslash \{u\}$;

$(x_i, y_i) \leftarrow$ Position generated by the direction u;

if $(x_i, y_i) \notin T$ **then**

$t \leftarrow false$;

$T[i] \leftarrow (x_i, y_i)$;

else if $(K = \emptyset)$ **then**

$t \leftarrow false$;

$i \leftarrow 2$;

end if

end while

end while

end

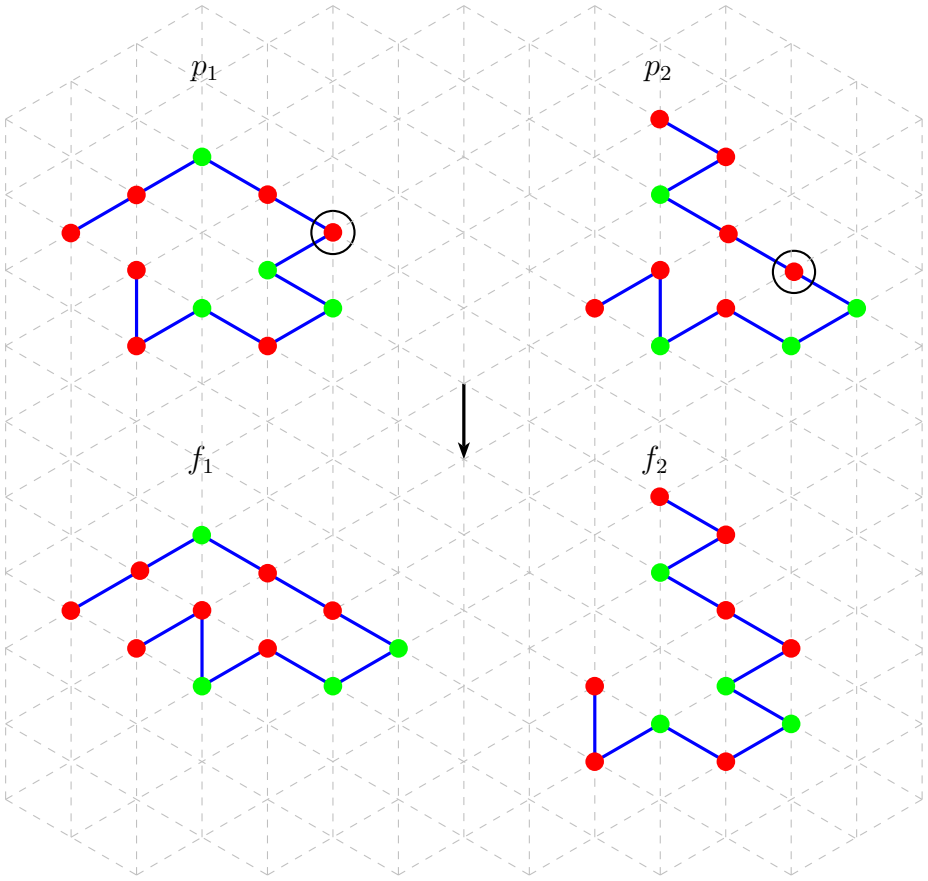

Figure 4: Crossover operator applied on two conformations of the sequence $H^2PH^2P^2HPH^2$. The circled nodes indicate the cutting points positions.

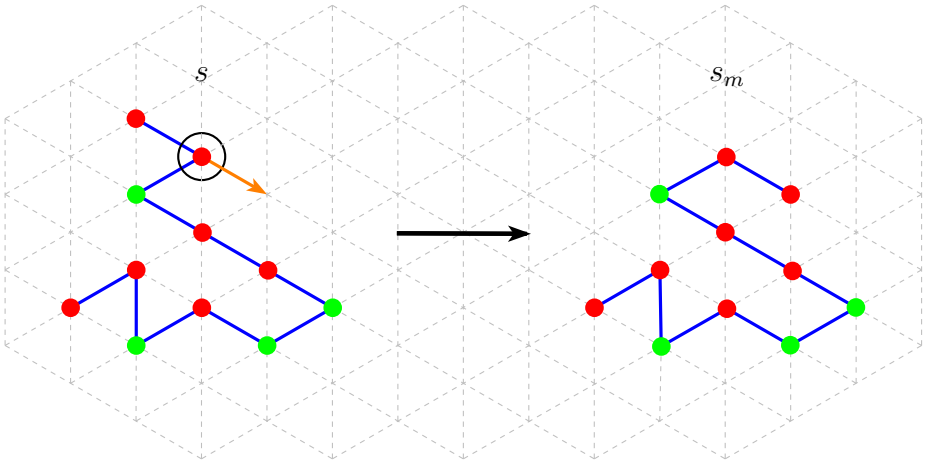

Figure 5: Mutation operator applied to the sequence $H^2PHP^2H^2PH^2$. The circled node indicate the position of the mutation point.

Algorithm 3 Local Search

Require: A feasible solution s, and its associated energy value $E(s)$.

Ensure: The best found solution s_{best}.

begin

Initialization:

$E(s_{best}) \leftarrow E(s)$;

$s_{best} \leftarrow s$;

while the stopping criterion is not checked **do**

$u \leftarrow$ random point of mutation;

$s \leftarrow$ mutation(s, u);

Evaluate s, i.e., calculate $E(s)$;

if $E(s) < E(s_{best})$ **then**

$s_{best} \leftarrow s$;

$E(s_{best}) \leftarrow E(s)$;

end if

end while

end

4.4 Selection Operator

It is a technique that favorites the best solutions to participate in the reproduction phase, in order to create new solutions with "satisfactory" quality. In this work we will use the roulette wheel selection technique (RWS). In this approach, each individual has a probability of being selected in accordance with his or her performance, so the more individuals are adapted to the problem, the more likely they are to be selected. The probability of selecting a solution i among the m solutions is given by the following:

$$p(i) = \frac{f(i)}{\sum_{j=1}^{m} f(j)}.$$

The RWS is an iterative operator, where in each step (after the probability assignment), it withdraws a random value form the range [0,1] (or [0,100], [0,360] depending on the representation of the selection wheel), and then it selects the corresponding individual. Table 1 presents an example of proportions assignment of four individuals according to their fitness evaluation.

Figure 6 presents the circular representation (i.e., pie chart) of the obtained selection probabilities in Table 1. As it is shown in Figure 6, the spinner indicates to select the chromosome I_2.

Individuals	Fitness: f_j	% of the total sum
I_1	3	7.5
I_2	12	30
I_3	5	12.5
I_4	20	50
Total sum	40	100

Table 1: Illustrative example of 4 solutions adduced by their fitness evaluations and the corresponding selection proportions.

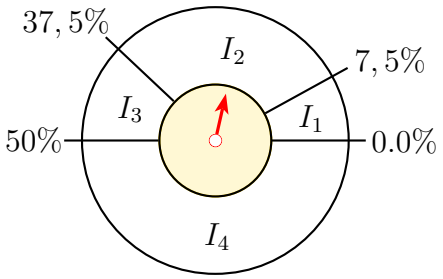

Figure 6: Roulette wheel selection.

4.5 Generating an offspring by the guided search strategy

For the reproduction phase, we propose an efficient algorithm that guides the search process to produce an offsprings of "good" quality. This algorithm combines the Local Search algorithm (LS) and the Tabu Search algorithm (TS). The role of the LS algorithm is to improve the quality of the solutions obtained by the crossover process, and in order to use the parent's information more effectively and to explore more research space; we propose to use the tabu search algorithm that memorizes the last points crossover used for some parents selected, see Algorithm 4. The working mechanisms of the suggested approach can be summarized by the following steps:

Step1 Generate n solutions for the initial population.

Step 2 Evaluate the fitness of each solution.

Step 3 Create a new population by the following steps:

Step3.1 Select two parents from the n solutions.

Step 3.2 Apply Algorithm 1, with a probability p_c.

Step 3.3 Apply Algorithm 2, with a probability p_m.

Step 3.4 Replace the previous population, with the current population.

Step 4 If the stopping criterion is not checked goto **Step 2**.

Algorithm 4 The reproduction algorithm guided by the local search algorithm and the tabu search strategy.

Require: Two selected parents p_1, p_2.

Ensure: Two individuals produced from parents, i.e., offsprings.

begin
Initialization:
$E_1 \leftarrow E(p_1)$;
$offspring_1^* \leftarrow p_1$;
$E_2 \leftarrow E(p_2)$;
$offspring_2^* \leftarrow p_2$;
$T \longleftarrow \emptyset$; // the tabu list
$K \leftarrow 0$;
while the stopping criterion is not checked **do**
$K \leftarrow K + 1$;
$u \leftarrow$ random point of crossover;
if $u \notin T$ **then**
$(offspring_1, offspring_2) \leftarrow$ crossover (p_1, p_2);
$u_1 \leftarrow$ random $[0, 1]$;
if $u_1 \leq p_m$ **then**
$offspring_1 \leftarrow$ local search $(offspring_1)$;
$offspring_2 \leftarrow$ local search $(offspring_2)$;
end if
if $(E(offspring_1) < E_1)$ **then**
$offspring_1^* \leftarrow offspring_1$;
$E_1 \leftarrow E(offspring_1)$;
end if
if $(E(offspring_2) < E_2)$ **then**
$offspring_2^* \leftarrow offspring_2$;
$E_2 \leftarrow E(offspring_2)$;
end if
end if
$T \longleftarrow T \cup \{u\}$;
end while
return $(offspring_1^*, offspring_2^*)$;
end

5 Experimental Results

The aim of this section is to assess the performance of the suggested approach. For the following experimental study, we used a several benchmark instances (i.e., protein sequences) presented in the H-P model of different sizes [8, 25]. Furthermore, we have selected the most studied in the literature instances to conduct the forthcoming experiments. Table 2 is composed of 3 columns, the first one represents the number of the protein sequence, the second represents the length of the protein sequence and the third represents the protein sequence in the H-P model, where the symbol $(.)^i$ means i fold repetitions of the respective subsequence in the brackets. For example, $(PH)^2$ is the simplified form of the sequence $PHPH$. The experimental results showed in Table 3 represents a comparison between the best results obtained by the proposed approach, and the above stated algorithms used to solve the PSP in the 2D triangular lattice model, namely HGA [10], TS [4], ERS-GA [23], HHGA [23], IMOG [26] for each instance given in Table 2. We show that all the approaches can provide an optimal confirmation when the length of the sequence is less than 36. However, the results obtained by our approach are better than the other approaches for the sequences 4, 6 and 7 as shown in Table 3 (see the respective conformation in Figure 9). However, The best prediction obtained by the algorithms GALSTS, TS , HHGA and IMOG are better than the HGA algorithm, ERS-GA and SGA for all sequences used for this experimental study.

Seq.	Length	Protein sequence in the H-P model
1	20	$(HP)^2PH(HP)^2(PH)^2HP(PH)^2$
2	24	$H^2P^2(HP^2)^6H^2$
3	25	$P^2HP^2(H^2P^4)^3H^2$
4	36	$P(P^2H^2)^2P^5H^5(H^2P^2)^2P^2H(HP^2)^2$
5	40	$P^2H(P^2H^2)^2P^5H^{10}P^6(H^2P^2)^2HP^2H^5$
6	50	$H^2(PH)^3PH^4PH(P^3H)^2P^4(HP^3)^2HPH^4(PH)^3PH^2$
7	60	$P(PH^3)^2H^5P^3H^{10}PHP^3H^{12}P^4H^6PH^2PHP$
8	64	$H^{12}(PH)^2((P^2H^2)^2P^2H)^3(PH)^2H^{11}$
9	85	$H^4P^4H^{12}P^6(H^{12}P^3)^3HP^2(H^2P^2)^2HPH$
10	100	$P^3H^2P^2H^4P^2H^3(PH^2)^3H^2P^8H^6P^2H^6P^9HPH^2PH^{11}P^2H^3PH^2PHP^2HPH^3P^6H^3$

Table 2: The used benchmark instances in the H-P model.

The experiment results, in Tables 3 and 4, represent a comparison between the best results obtained by GALSTS, with some existing results based on several large benchmarks found in the literature, solved by HGA and SGA [10], TS [4], ERS-GA [23], HHGA [23], and IMOG [26] approaches. Clearly, the number of possible conformations increases exponentially when the size of the instance increases. According to their energy values, all approaches could provide an optimal confirmation when the size of the instance is less than 36. But, as it is shown in Tables 3 and 4, the best conformations obtained by

GALSTS are better than all the cited approaches bellow, based on the instances of size up to 36. This demonstrates the ability of GALSTS to explore the search space more effectively comparing with the other approaches.

Seq.	Length	SGA	HGA	TS	ERS-GA	HHGA	IMOG	GALSTS	Conformation
1	20	-11	**-15**	**-15**	**-15**	**-15**	**-15**	**-15**	Fig7(a)
2	24	-10	-13	**-17**	-13	**-17**	**-17**	**-17**	Fig.7(b)
3	25	-10	-10	**-12**	-12	**-12**	**-12**	**-12**	Fig.7(c)
4	36	-16	-19	**-24**	-20	-23	**-24**	**-24**	Fig.7(d)
5	48	-26	-32	-40	-32	-41	-40	**-43**	Fig.7(e)
6	50	-21	-23	NA	-30	-38	**-40**	**-40**	Fig.7(f)
7	60	-40	-46	**-70**	-55	-66	-67	**-70**	Fig.7(g)
8	64	-33	-46	**-50**	-47	-63	-63	**-67**	Fig.7(h)
9	85	NA	NA	NA	NA	NA	NA	**-98**	Fig.7(i)
10	100	NA	NA	NA	NA	NA	NA	**-87**	Fig.7(j)

Values in bold indicate the best obtained evaluation for the correspondent instance.
NA refers to not available data in literature.

Table 3: The best conformations obtained by GALSTS compared with other algorithms for 10 H-P sequences in 2D triangular lattice model.

Instances larger than 64 are not covered in the literature for the 2D triangular lattice model. However, they were processed for the rectangular model [27]. According to the obtained results in Table 3, a strong improvement in energy appears clearly compared with the triangular model. Table 4, graphically presented in Figure 7 and Figure 8, show a performance comparison on the stability of our approach and three other algorithms HHGA, IMOG and ERS-GA, such that the efficiency of the algorithms is measured by the best and means results in 30 independent runs for each sequence. We show that for the most instances, the proposed approach is able to find the best optimal solutions and achieves a better average solution quality than other algorithms. the average solution quality is very encouraging.

		ERS-GA		HHGA		IMOG		GALSTS	
Seq.	Length	Best	Mean	Best	Mean	Best	Mean	Best	Mean
1	20	**-15**	-12.50	**-15**	-14.73	**-15**	-14.73	**-15**	**-14.86**
2	24	-13	-10.20	**-17**	-14.93	**-17**	-14.93	**-17**	**-15.53**
3	25	12	-8.47	**-12**	- 11.57	**-12**	-11.57	**-12**	**-12**
4	36	-20	-16.17	-23	-21.27	-23	-21.27	**-24**	**-21.93**
5	48	-32	-28.13	-41	- 37.30	-41	-37.30	**-43**	**-39.86**
6	50	-30	-25.30	-38	-34.10	-38	-34.10	**-40**	**-37.6**
7	60	-55	-49.43	-66	- 61.83	-66	-61.83	**-70**	**-68.26**
8	64	-47	-42.37	-63	- 56.53	-63	-56.53	**-67**	**-58.46**

Values in bold indicate the best obtained evaluation for the correspondent instance.

Table 4: A comparative study on the stability and the best prediction of the GALSTS with other algorithms.

Table 5 resumes the obtained results for 30 independent runs per each of the above stated instances. The aim of this experiment is to compare the suggested algorithm GALSTS against two competing algorithms ERS-GA and SGA. The results are adduced according to the best and worst overall evaluation and their corresponding deviation from the best known value (BKV). The proposed algorithm GALSTS is shown to be more effective than the other competing algorithms; even when comparing its worst produced conformation to their best ones, with a sole exception of the first tested sequence, where it shows a slight difference. However, when increasing the size of the instance, it is clear that the suggested algorithm is incrementally taking advantage over the competing algorithms, even when comparing its worst solution to their best ones. Furthermore, the suggested algorithm is shown to be able to attain good quality conformations or even optimal, with a sole exception the sixth tested instance, where it shows a one unit deviation of the best known evaluation.

			GALSTS		ERS-GA	SGA
Seq.	length	E^*	Best (dev. BKV)	Worst	Best (dev. BKV)	Best (dev. BKV)
1	20	-15	**-15** (**00**)	-14	**-15** (**00**)	-11 (04)
2	24	-17	**-17** (**00**)	-15	-13 (04)	-10 (07)
3	25	-12	**-12** (**00**)	-12	**-12** (**00**)	-10 (02)
4	36	-24	**-24** (**00**)	-21	-20 (04)	-16 (08)
5	48	-43	**-43** (**00**)	-38	-32 (11)	-26 (17)
6	50	-41	**-40** (**01**)	-36	-30 (11)	-21 (20)
7	60	-	**-70** (-)	-65	-55 (-)	-40 (-)
8	64	-	**-67** (-)	-56	-47 (-)	-33 (-)

Values in bold indicate the best obtained evaluation for the correspondent instance.

Table 5: Comparison between the results, the best and worst evaluations obtained by GALSTS and the best results of SGA and ERS-GA. E^* is the best energy value.

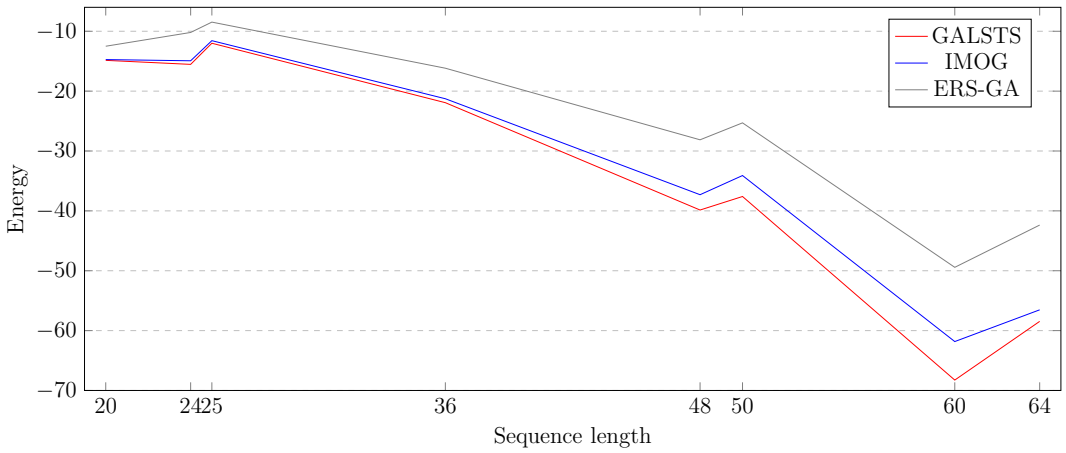

Figure 7: Illustration of the comparison results regarding the mean energy obtained using GALSTS against other algorithms.

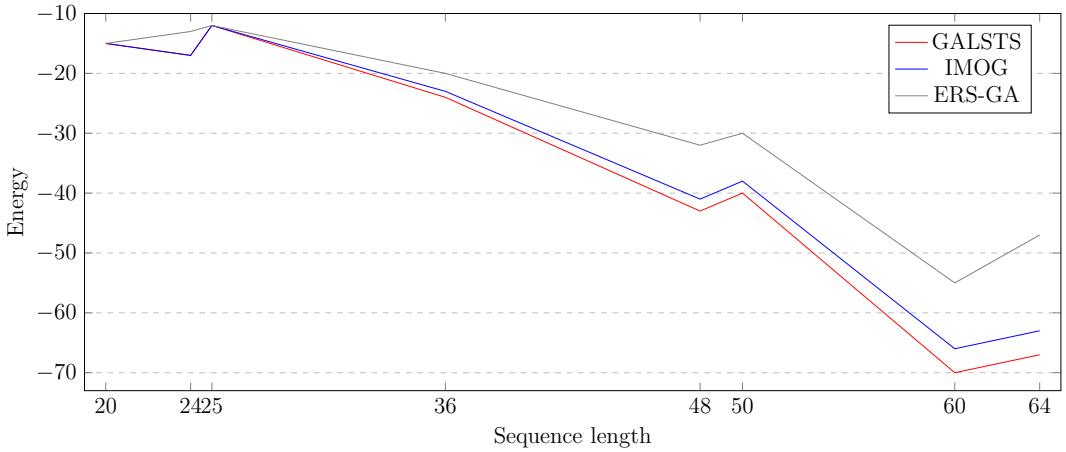

Figure 8: Illustration of the comparison results regarding the best predictions obtained using GALSTS against other algorithms.

6 Conclusions and Perspectives

This paper presents an efficient approach called GALSTS for solving the protein structure prediction in 2D triangular model using the simplified hydrophobic-polar model. An initialization algorithm was proposed that allows to generate only valid conformations for the initial population of GALSTS. This algorithm eliminates cyclic movements during the construction of solutions. GALSTS consists in using Tabu and Local Search algorithm to explore the search space handling more efficiently. This approach allows to use the information provided by the selected parents to produce solutions with good quality. From our experimental results, GALSTS was able to find the best known solutions and it is more effective for the stability results than other existing algorithms. In terms of future scope of applications, GALSTS can be used to solve the PSP problem in the 3D cubic and 3D triangular models, it can also be used to solve other optimization problems in the combinatorial optimization framework.

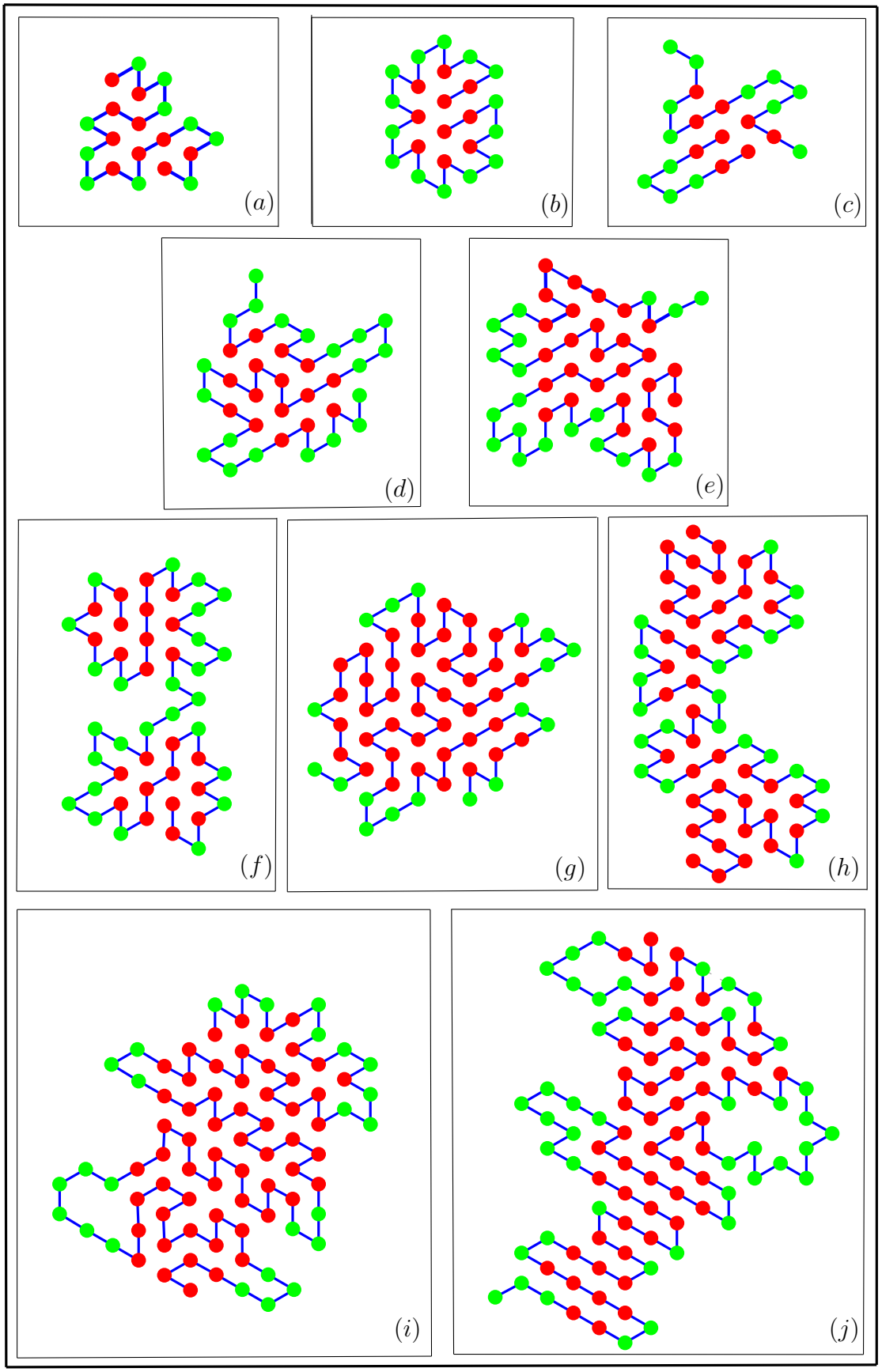

Figure 9: (a) to (j) Results of the best conformation structure of ten protein sequences.

References

[1] Richa Agarwala, Serafim Batzoglou, Vlado DanÄik, Scott E. Decatur, Sridhar Hannenhalli, Martin Farach, S. Muthukrishnan, and Steven Skiena. Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP model. *Journal of Computational Biology*, 4(3):275–296, 1997.

[2] Bonnie Berger and Tom Leighton. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *Journal of Computational Biology*, 5(1):27–40, 1998.

[3] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3):268–308, 2003.

[4] Hans-Joachim Bockenhauer, Abu Zafer M. Dayem Ullah, Leonidas Kapsokalivas, and Kathleen Steinhofel. A local move set for protein folding in triangular lattice models. In *International Workshop on Algorithms in Bioinformatics*, pages 369–381. Springer, 2008.

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

[6] 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.

[7] Vincenzo Cutello, Giuseppe Nicosia, Mario Pavone, and Jonathan Timmis. An immune algorithm for protein structure prediction on lattice models. *IEEE transactions on evolutionary computation*, 11(1):101–117, 2007.

[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] Ken A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24(6):1501–1509, 1985.

[10] Md Tamjidul Hoque, Madhu Chetty, and Laurence S. Dooley. A hybrid genetic algorithm for 2d FCC hydrophobic-hydrophilic lattice model to predict protein folding. In *Australasian Joint Conference on Artificial Intelligence*, pages 867–876. Springer, 2006.

[11] Md Kamrul Islam and Madhu Chetty. Clustered memetic algorithm with local heuristics for ab initio protein structure prediction. *IEEE Transactions on Evolutionary Computation*, 17(4):558–576, 2013.

[12] Tianzi Jiang, Qinghua Cui, Guihua Shi, and Songde Ma. Protein folding simulations of the hydrophobicâhydrophilic model by combining tabu search with genetic algorithms. *The Journal of chemical physics*, 119(8):4592–4596, 2003.

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

[14] Natalio Krasnogor, B. P. Blackburne, Edmund K. Burke, and Jonathan D. Hirst. Multimeme algorithms for protein structure prediction. In *International Conference on Parallel Problem Solving from Nature*, pages 769–778. Springer, 2002.

[15] Natalio Krasnogor, William E. Hart, Jim Smith, and David A. Pelta. Protein structure prediction with evolutionary algorithms. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2*, pages 1596–1601. Morgan Kaufmann Publishers Inc., 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] 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.

[18] David A. Pelta and Natalio Krasnogor. Multimeme algorithms using fuzzy logic based memes for protein structure prediction. In *Recent advances in memetic algorithms*, pages 49–64. Springer, 2005.

[19] GÃ$\frac{1}{4}$nther R. Raidl, Jakob Puchinger, and Christian Blum. Metaheuristic hybrids. In *Handbook of metaheuristics*, pages 469–496. Springer, 2010.

[20] 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.

[21] 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.

[22] Alena Shmygelska and Holger H. Hoos. An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem. *BMC bioinformatics*, 6(1):30, 2005.

[23] Shih-Chieh Su, Cheng-Jian Lin, and Chuan-Kang Ting. An efficient hybrid of hill-climbing and genetic algorithm for 2d triangular protein structure prediction. In *2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, pages 51–56. IEEE, 2010.

[24] El-Ghazali Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009.

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

[26] Cheng-Hong Yang, Kuo-Chuan Wu, Yu-Shiun Lin, Li-Yeh Chuang, and Hsueh-Wei Chang. Protein folding prediction in the HP model using ions motion optimization with a greedy algorithm. *BioData mining*, 11(1):17, 2018.

[27] Xinchao Zhao. Advances on protein folding simulations based on the lattice HP models with natural computing. *Applied Soft Computing*, 8(2):1029–1040, 2008.