Available online at http://scik.org
Commun. Math. Biol. Neurosci. 2023, 2023:117
https://doi.org/10.28919/cmbn/8071
ISSN: 2052-2541

AN INTEGER PROGRAMMING MODEL FOR PREDICTING MULTI-SHAPES OF 3D PROTEIN STRUCTURE MODEL

ALAA FAHIM[1,*], NEHAD ABDELRAHEEM[2]

[1]Mathematics Department, faculty of science, Assuit University, Egypt

[2]Faculty of Computers and Information, Assuit University, Egypt

Abstract. Protein structure prediction is using bioinformatics and computational science tools is a vital part of medication discovery and infection expectation, but calculating the structure of a folded protein is a difficult task. In this paper, we predict protein structure of hydrophobic-polar lattice models in three dimensions. The considered problem is treated as an integer programming problem. The model finds multiple optimal shapes with the same minimal energy of the protein. Optimization is performed by Tabu search and the sequences are investigated by blending two fundamental techniques: strengthening and enhancement. The effectiveness of our strategy was confirmed in comparison with existing approaches benchmark protein sequences.

Keywords: Tabu search; mathematical model; 3D HP model.

2020 AMS Subject Classification: 92C40.

1. INTRODUCTION

Protein folding [1] is an interesting topic and it is picked by Given an amino-acid sequence, Protein Structure Prediction (PSP) seeks the optimal form that minimize the energy of the folded protein. Combining bioinformatics and computational science, the PSP problem is integral part

*Corresponding author

E-mail address: alaa@aun.edu.eg

Received June 27, 2023

of drug discovery and disease expectations. Accordingly, the PSP problem has attracted much attention over the past 30 years.

Chan [2] presented a visualization method of PSP and demonstrated its superiority over other examined techniques in protein folding. They investigated the hyrophobic-polar (HP) model [3], which divides the 20 amino acids into two classes of residue: hydrophobic (nonpolar) and hydrophilic (polar). Protein information is usually acquired by X-ray crystallography and nuclear magnetic resonance, but these methods are costly and time consuming.

The three- dimenional (3D) structures of HP proteins have been predicted by various algorithms, including (but not limited to) memetic Algorithm [4], GA (Genetic Algorithms)[5], TS (Tabu Search)[5]. Each strategy captures the protein structure in a characteristic way. For example, GA perform biologically inspired operations such as mutation, mutation, crossover, and selection, whereas TS attempt to balance the intensification and diversification of the search procedures. To this end, TS alter its decision rules to empower the best move mixes and arrangement. As another procedure in 3D PSP problem, we introduced a mathematical formulation of the integer programming problem. Mathematical treatments of HP models have been rarely attempted owing to the difficulty of the establishing the integer programming formulation of 3D HP models with side chains [6]. The present paper, develop a simple, understandable, and time-saving integer programming for the PSP problem.

The integer programming problem managed by TS is called HP-TS. TS [7] uses a versatile memory and conduct a responsive investigation; that balance diversification against intensification. The diversification methodology relies upon the neighborhood structure of the TS technique. The TS tree covers most districts in the search space and maintains an assorted variety of sequences. Furthermore, a Tabu List (TL) widens the interest of the search in unvisited regions of the solution space, preventing trapping of the solution in local minima. To refine the currently best solutions, our strategy adds a pattern search method at the last stage. We also execute two techniques Attract H strategies that control the movements of some promising H nodes to accelerate the procedure toward the best solution.

The reminder of the paper is organized as follows. Section 2 (”HP Lattice Model”) introduces some principle definitions related to the PSP problem. Section 3 introduces our HP model

reformulated as a whole integer programming problem model. Section 4 presents the TS and implementation of the HP model. Section 5 discuss the consequences of the HP-TS strategy and compare performance of the proposed technique and other strategies. Conclusions are drawn in Section 6.

2. PRELIMINARIES

This section highlights the basic elements, and background of our proposed methods.

2.1. HP Lattice Model. Protein consist of 20 amino acids divided into two classes depending on their hydrophobicity: hydrophobic (H) and polar (P) as mentioned above. Hydrophobic amino acids tend to aggregate while excluding water to minimize the energy of the protein structure. The lattice model configures a sequence of H and P amino acids and determines the shape of the protein's conformation.

The HP model is the most thoroughly examined strategy for evaluating protein structure (see Figure 1). HP models can determine the compliance of the genuine protein, the local minimal structure, and the global minimal structure. These features reflect how the protein adapts to avoid the internal hydrophobic monomers.

The 3D lattice HP model has been solved by metaheuristics such as like GA [8, 9, 10], memetic algorithms [4], evolutionary methods [11], ant colony optimization [12], and an improved TS method [13, 14]. Mathematical models of our problem have been proposed in few papers [15, 16], but our formulation is simpler than than these methods, and finds a more accurate , solution within a shorter runtime. Our method also finds multiple shapes that optimize the protein structure.

2.1.1. *Protein Structure.* Our PSP method proceeds in three steps as follows.

- A general; protein is written as $S = \{s_1, ..., s_n\}$, with $s_i = \{H, P\}$, where n denotes the number of amino acids in the chain and S vector explores the arrangement of the H and P monomers.
- To explore the search space, we define a direction vector X of length n_2. the forward, leftward, rightward, upward, and downward directions by 0, 1, 2, 3, and 4, respectively.

- Coordinate every node in HP model with (x,y,z) coordinatesof every node in the HP model are stored in matrix M. The coordinate of the first two mers are $(0,0,0)$, and $(1,0,0)$, respectively.

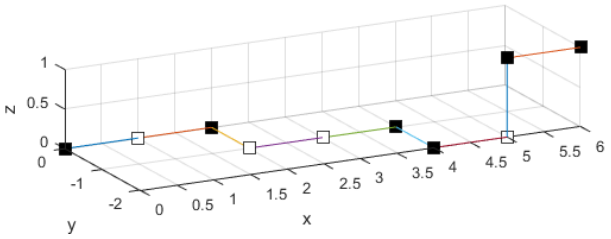


FIGURE 1. Example of an HP lattice model

For example, the HP model in Figure 1 shows that the black and white nodes denote H and P monomers, respectively. Lattice model in Figure 1 can be expressed as,

- $S = \{HPPHPPHPHPPHP\}$,
- Direction vector $X = \{4,4,1,2,2,3,2,0,3,3,2\}$,
- M matrix of coordinate nodes (see Table 1).

TABLE 1. Coordinate nodes in HP model

M_X	0	1	1	0	0	−1	−1	−1	0	1	1	0	0
M_y	0	0	0	0	1	1	0	−1	−1	−1	−1	−1	−2
M_z	0	0	−1	−1	−1	−1	−1	−1	−1	−1	0	0	0

3. INTEGER PROGRAMMING FORMULATION OF THE PSP PROBLEM

The PSP problem can be written as the following integer programming problem:

$$\max \quad \sum_{l,k} y_{lk},$$

$$where \quad y_{lk} = \begin{cases} 1, & if \|M_l - M_k\| = 1 \\ 0, & \text{others.} \end{cases}$$

Where $l = \{1,\ldots,n-2\}$ and $k = \{l+2,\ldots,n\}$.

The problem is subjected to three constraints: an overlapping constraint that forbids overlap of any two mers in the protein structure, a connectivity constraint is to the connectivity of the

protein after one mer move, and the boundary constraint that limits the length of the protein thus avoiding a straight graph of HP lattice.

- **Overlapping constraint**

$$\|M_i - M_j\| \geq 1.$$

where $i = \{1, \ldots, n-1\}$ and $j = \{i+1, \ldots, n\}$.

- **Connectivity constraint**

$$\|M_i - M_{i+1}\| = 1.$$

where $i = \{1, \ldots, n-1\}$.

- **Bounding constraint**

$$length(X) < Pbound,$$

$$length(Y) < Pbound,$$

$$length(Z) < Pbound$$

where $Pbound = n/3$; and $length(X)$, $length(Y)$ and $length(Z)$ are the lengths of the HP model in the x,y, and z direction, respectively.

Using, the penalty methodology [17], we transformed this constrained problem into a series of unconstrained problems. The unconstrained solutions of these problems should converge to those of the constrained problem.

4. TABU SEARCH

The TS is a heuristic methodology proposed by Glover [18] that solves combinatorial optimization problems using adaptive memory highlights[18, 19, 20]. With its TL, adaptive memory enables the search process, prevents trapping in local optimal solutions, and successfully explores the solution space. The TL includes permitted neighboring solutions and excludes solutions that fail the conditions. A tabu method begins with an initial solution X, creates neighbors, and moves to the nearest available neighboring solution $N(X)$. The best solution is selected as a filter solution X_c. If X_c satisfies the ambition rule, it supplants the current solution X and is added to TL T_{list}; otherwise, the present solution X is supplanted by the current

best solution X', $(E(X) = min\{E(X) \mid X \in N(X), X \in T_{list}\})$ and Tabu list will contain X'. In general,T_{list} is the first-in-first-out memory with limited length, meaning that solutions found in the early stage might not be searched. at later staged T_{list} is constructed using the simple descend method. This technique allows moves only to neighbor solutions that improve the current value of the objective function. In TS procedures that incorporate longer-term considerations, N(x) may, include non-customary solutions such as those found and assessed in past pursuits, or those distinguished as high-quality neighbors of these past solutions. In this context, TS can be viewed as a unique neighborhood strategy. in which the area of x is not a static, but can change depending on the search history N(x) then becomes a modified neighborhood.

4.1. Basic Data Structure. The (x,y,z) coordinates of each node in the lattice model are included in the M matrix. In addition, the data of the nodes adjacent to each node in the sixth direction upward, downward, leftward, rightward, forward, and backward directions are stored in a matrix called UDLRFB matrix.

The UDLRFB matrix reveals the locations around each node. The locations around the nodes in Figure 1 are given in Table 2.

TABLE 2. Basic data sturcture of the nodes in Figure 1

Nodes	M	UDLRFB Matrix					
S	coordinate	Upward	Downward	Leftward	Rightward	Forward	Backward
H	$(0,0,0)$	0	12	2	0	0	4
P	$(1,0,0)$	0	11	0	1	0	3
P	$(1,0,-1)$	0	10	0	4	2	0
H	$(0,0,-1)$	5	9	3	7	1	0
P	$(0,1,-1)$	0	4	0	6	0	0
P	$(-1,1,-1)$	0	7	0	9	0	0
H	$(-1,0,-1)$	6	8	4	0	0	0
P	$(-1,-1,-1)$	7	0	8	0	0	0
H	$(0,-1,-1)$	4	0	10	8	12	0
P	$(1,-1,-1)$	3	0	0	9	11	0
P	$(1,-1,0)$	2	0	0	12	0	10
H	$(0,-1,0)$	1	13	11	0	0	9
P	$(0,-2,0)$	12	0	0	0	0	0

4.2. Initial Solution. The initial solution X is generated by selecting random values of length n-1, where n is the sequence length of the lattice, The values range from 0 to 4. For example, suppose that X=$\{0, 1, 2, 3, 1, 4, 2,..., 4\}$. The method is outlined in Procedure 4.1 below.

Procedure 4.1. Initial Solution

1. Initialize the coordinates of the first and second nodes as (0,0,0) and (1,0,0) respectively.

2. For $i = 1$ *to* $n-1$ *do Step 3.*

3. Take value X_i *from 0 to 4 following a normal distribution.*

4.3. Neighborhood Local Structure. The main task of TS is generating the neighborhood structure. In our method, the neighborhood structure is generated by the following steps.

4.3.1. *Random Direction Change.* The Random Direction Change (RDC) diversifies the search process. One or more entries X_i are selected from the direction vector X and their values are changed. For example, if $X_i = 0$ (i.e. X_i poised to move in the forward direction), its new value is selected from $\{0,1,2,3,\ldots,4\}$.

Procedure 4.2. *RDC*

1. For $i = 1$ *to* num-nodes *do Steps 2 and 3.*

2. Select a random number r from $1,\ldots,n-2$.

3. Take a value of X_r *from* $[0,4]$.

4.3.2. *Construct Tree of Neighbourhood.* A tree of the neighborhood is constructed on two main levels. In level 1, the RDC method (Procedure 4.2) generates *Ntrial* solutions to increase the diversity of the current solutions. In level 2, the RDC method generates η solution from every previous solution and selects best *Ntrial* solution among all available solutions. The neighborhood construction is outlined in Procedure 4.3, and a generated neighborhood tree is presented in Figure 2.

Procedure 4.3. *Construct a tree of the neighborhood*

1. Generate one solution by Procedure 4.1.

2. Generate Ntrial *solutions by the RDC method (Procedure 4.2).*

3. Sort all available solutions and select the best Ntrial *solution.*

4. Generate η solutions for every previous solutions by theRDC method and repeat Step 3.

5. Repeat Steps 2 to 4 until the termination condition id satisfied.

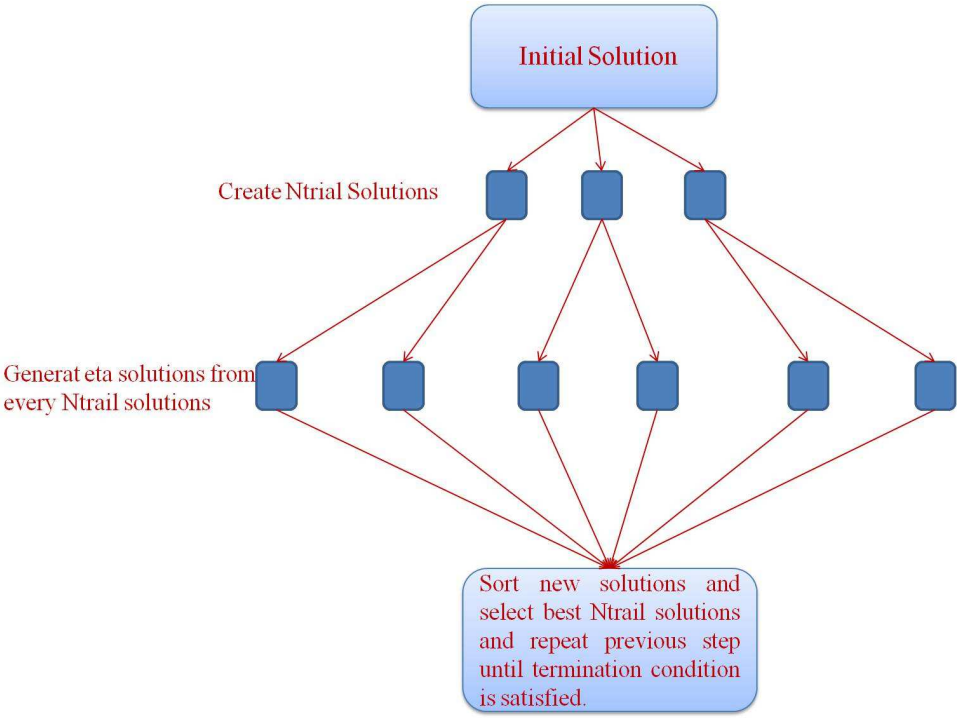
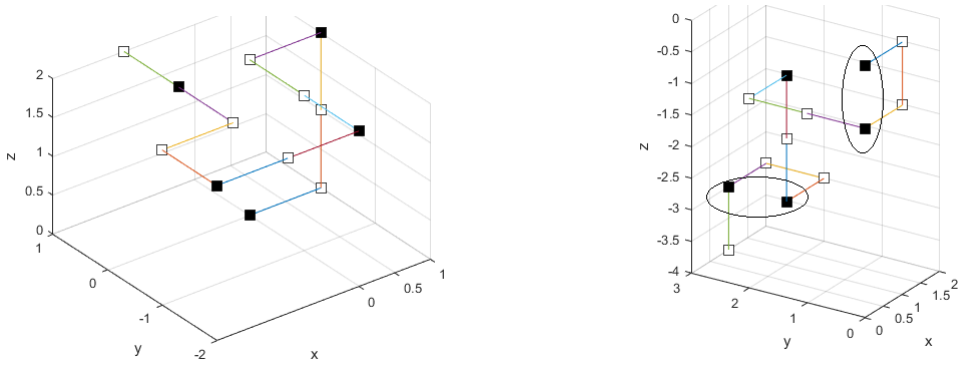


FIGURE 2. Tree of the neighbourhood structure

4.4. Intensification. The intensification process is important for extracting the most elite solution because it encourages searching of the most promising localities in the search space. In the protein folding problem, intensification assist the Attract H method, describe below.

4.4.1. *Attract H Method.* This process guides the locations of the H nodes by moving each H node toward other H nodes, as near as possible. This movement should minimize the energy of the current protein. The procedure is given as Procedure 4.4.



(A) (B)

FIGURE 3. A simple HP model before (a) and after (b) applying the attract H algorithm

Figure 3 shows the effect of the Attract H method on the HP model sequence, Panels a and b of this Figure display the HP model before and after applying the respectively. Note that Attract H method increases the number of H-H separated nodes.

Procedure 4.4. *Attract H*

1. *Detect all H nodes with at most one umbound neighborhood H node and insert them into the HH vector.*
2. *Detect the free locations from the UDLRFB matrix to the first previously detected node.*
3. *For i=1 to number of H nodes do steps 4 - 6.*
4. *Select a new location for H among the possible locations in the H nodes neighborhood.*
5. *Change coordinates of all remaining nodes from S_{ri+1} to S_n until connectivity is achieved.*
6. *If the solution overlaps another solution or expands the H-H distances, it is discarded.*

4.5. HP-TS Algorithm. The HP-TS Algorithm is outlined in Figure 4, After initializing the TL. the algorithm generate the neighborhood solutions applies the intensification method, and updates the TL. The algorithm iterates until the termination condition is satisfied.

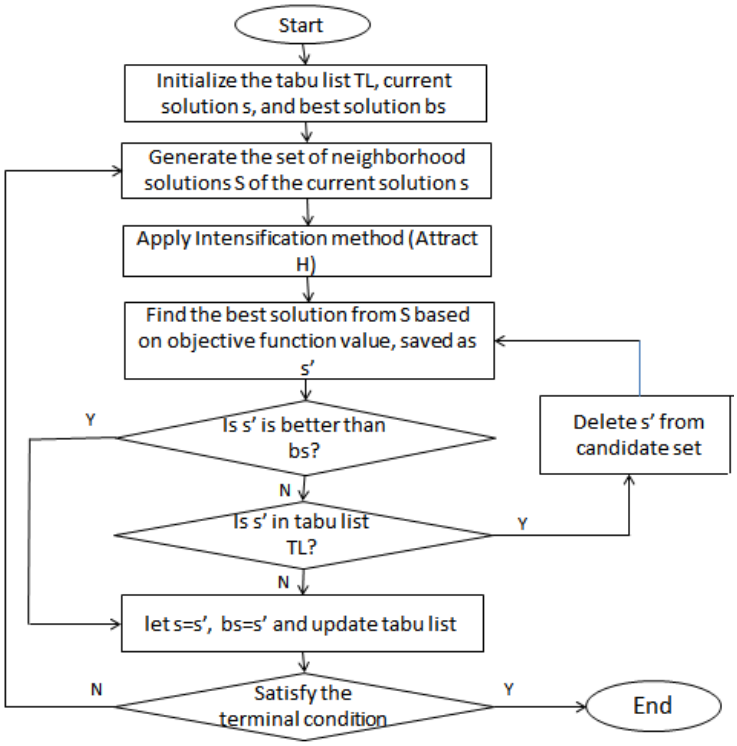


FIGURE 4. HP-TS Algorithm

5. NUMERICAL EXPERIMENTS

This section compares the results *HP-TS* and some existing models on 14 HP benchmark models.

5.1. Parameters Setting. All parameters were assigned their most common values in the literature or were valued in our preliminary numerical experiments. The values are listed below

- **TS Operator Parameters**
  - **-:** Length of TL :$TLmax = 50$.
  - **-:** Length of best solutions list: $TBmax = 5$.
  - **-:** Maximum number of changed nodes on a zone : $max_{zone} = 10$.
  - **-:** Tree neighborhood parameters, $Ntrial = 10, \eta = 10$.
- **Penalty Parameters**
  - **-:** $mu = 1000$, is the penalty parameter.
  - **-:** $eps = 1e-5$, is the penalty parameter.
- **termination parameter**

-: *HP-TS* algorithm is forced to be terminated when the number of generations exceeds $t_{\max} = 2000$.

Performance Analysis. The *HP-TS* method was programmed in MATLAB and its performance of *HP-TS* method was tested on 14 benchmark HP problems (see Table 3).

No.	length	protein sequence
P1	5	HPPHP
P2	8	PHPHPPHP
P3	13	HPPHPPHPHPPHP
P4	17	HHHHPPHHHHHHHPPPH
P5	20	HPHPPHHPHPPHPHHPPHPH
P6	21	PHPHPPHPHPPHPPHPHPPHP
P7	24	HHPPHPPHPPHPPHPPHPPHPPHH
P8	25	PPHPPHHPPPPHHPPPPHHPPPPHH
P9	27	HHHHPPPPPHPPPPPHHHPPPPPPPPH
P10	34	HPPHPPHPHPPHPPHPHPPHPHPPHPPHPHPPHP
P11	36	PPPHHPPHHPPPPPHHHHHHHPPHHPPPPHHPPHPP
P12	48	PPHPPHHPPHHPPPPPHHHHHHHHHHPPPPPPHHPPHHPPHPPHHHHH
P13	50	HHPHPHPHPHHHHPHPPPHPPPHPPPPHPPPHPPPHPHHHHPHPHPHPHH
P14	60	PPHHHPHHHHHHHHPPPHHHHHHHHHHPHPPPHHHHHHHHHHHHPPPPHHHHHHPHHPHP

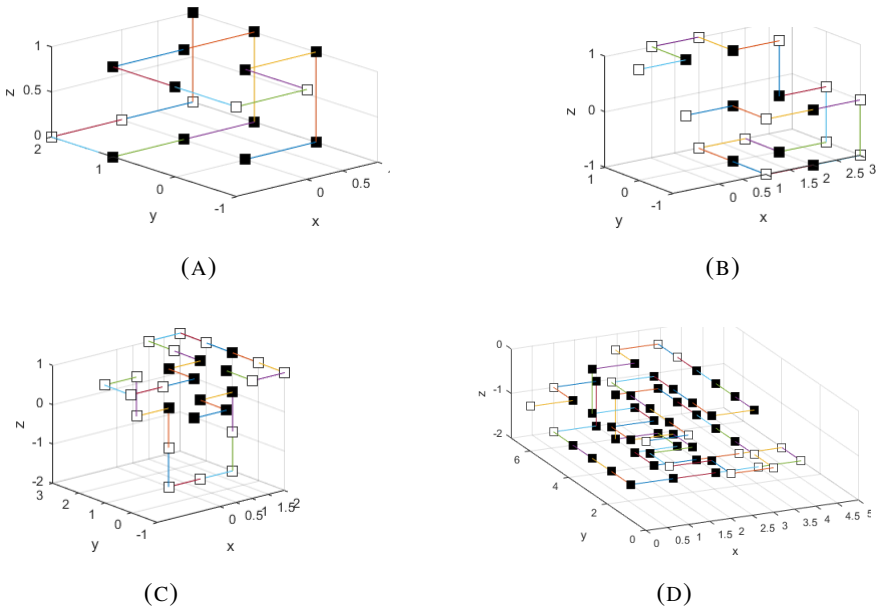
TABLE 3. Benchmark problems for performance evolution

As confirmed in Table 3, our proposed HP-TS method predicted of the folding of proteins with different lengths (from short to long length). Table 4 presents best energy values determined in in one run. The *HP-TS* method found the best-known solution in all HP problems and surpassed the best-known result on the P6 and P9 benchmarks.

TABLE 4. HP-TS results on the 14 benchmark problems

HP	P1	P2	P3	P4	P5	P6	P7
length	5	8	13	17	20	21	24
best-sol	–1	–2	–5	–9	–11	–8	–13
HP-TS	–1	–2	–5	–9	–11	**–9**	–13
HP	P8	P9	P10	P11	P12	P13	P14
length	25	27	34	36	48	50	60
best-sol	–9	–9	–19	–18	–29	–26	–49
HP-TS	–9	**–10**	–19	–18	–29	–26	–49

Figure 5 shows the lowest-energy structures of the benchmark proteins P4, P7, P9 and P12 (with residue lengths of 17, 21, 27 and 48 respectively) obtained by the HP-TS algorithm.



(A) (B)

(C) (D)

FIGURE 5. Conformation of sequences P4(a), P7(b), P9(c), and P12(d) obtained by the proposed method

The *HP-TS* method is strengthened by its ability to find more than one optimal conformation of the same protein model. Figure 6 presents the multi shapes of the P3 and P4 found by a proposed algorithm. Here we have selected small length proteins to clarify the structures. The

energy of both structures of sequence P3 (with 13 residue) was -5, (Figure 6a and 6b), and that of both structures of sequence P4 (with 17 residue) was -9.

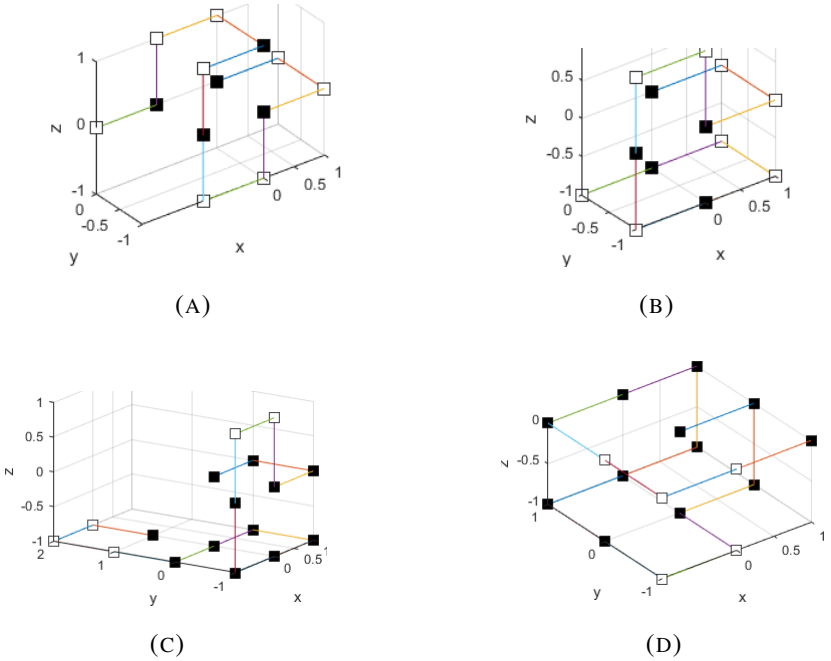


FIGURE 6. Two conformation of (a) and (b) of 13-length sequence P3 with energy -5 and (c) and (d) the 17-length sequence P4 with energy is -9

5.2. Comparison Results. The strength of the *HP-TS* method was verified in comparisons with other methods. Table 5 presents optimal results of our *HP-TS* method. the multiple crossover and mutation TS algorithm (MCMPSO-TS) [21], the hybrid GA and particle swarm optimization (PSO) algorithm (HGA-PSO) [6] and the two-phase PSO (TPPSO) methods [16] methods. As the MCMPSO-TS method was designed for small HP lengths, it evoluated on P1, P2, P3, P4, P5, P6, P8, P10, and P11 benchmarks. The HGA-PSO method was tested on the P5, P7, P8, P11, P12, P13, and P14 benchmarks, and the TPPSO method was tested only on P9, and P11. The*HP-TS* method covered all benchmark models with different lengths. The *HP-TS* method found the optimal solution to all models and outperformed the existing methods on P6 and P9.

HP	Length	best	MCMPSO-TS	HGA-PSO	TPPSO	HP-TS
P1	5	–1	–1	-	-	–1
P2	8	–2	–2	-	-	–2
P3	13	–5	–5	-	-	–5
P4	17	–9	–9	-	-	–9
P5	20	–11	–11	–11	-	–11
P6	21	-8	-8	-	-	**–9**
P7	24	–13	-	–13	-	–13
P8	25	–9	–9	–9	-	–9
P9	27	–9	-	-	–9	**–10**
P10	34	–19	–19	-	-	–19
P11	36	–18	–18	–18	–17	–18
P12	48	–29	-	–29	-	–29
P13	50	–26	-	–26	-	–26
P14	60	–49	-	–49	-	–49

TABLE 5. Performance comparison of HP-TS and other methods on the benchmark problems (the best solutions are shown in bold font)

6. CONCLUSION

To solve the PSP problem, we proposed an accurate and fast-running integer programming model (the *HP-TS* model) that combines diversification and intensification in local searching. The model include a procedure that aggregates the hydrophobic amino-acid residue, thus enhancing the diversification and intensification strategies. A performance comparison between the HP-TS and existing methods demonstrated that our method minimized the energies of proteins of different lengths, and surpassed the existing methods on two of the benchmarks proteins. Our proposed method obtains multiple folding shaped with the same minimal energy for the same protein sequence, which will benefit biological search.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] M. Paterson, T. Przytycka, On the complexity of string folding, in: F. Meyer, B. Monien (Eds.), Automata, Languages and Programming, Springer Berlin Heidelberg, Berlin, Heidelberg, 1996: pp. 658–669. https://doi.org/10.1007/3-540-61440-0_167.

[2] H.S. Chan, K.A. Dill, The protein folding problem, Phys. Today, 46 (1993), 24–32.

[3] K.A. Dill, Theory for the folding and stability of globular proteins, Biochemistry. 24 (1985), 1501–1509. https://doi.org/10.1021/bi00327a032.

[4] A. Bazzoli, A.G.B. Tettamanzi, A memetic algorithm for protein structure prediction in a 3D-lattice HP model, in: G.R. Raidl, S. Cagnoni, J. Branke, et al. (Eds.), Applications of Evolutionary Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004: pp. 1–10. https://doi.org/10.1007/978-3-540-24653-4_1.

[5] X. Lin, X. Zhang, F. Zhou, Protein structure prediction with local adjust tabu search algorithm, BMC Bioinform. 15 (2014), S1. https://doi.org/10.1186/1471-2105-15-s15-s1.

[6] C.J. Lin, S.C. Su, Protein 3d hp model folding simulation using a hybrid of genetic algorithm and particle swarm optimization, Int. J. Fuzzy Syst. 13 (2011), 140–147.

[7] A.R. Hedar, M. Fukushima, Tabu Search directed by direct search methods for nonlinear global optimization, Eur. J. Oper. Res. 170 (2006), 329–349. https://doi.org/10.1016/j.ejor.2004.05.033.

[8] M.T. Hoque, M. Chetty, A. Sattar, Protein folding prediction in 3D FCC HP lattice model using genetic algorithm, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, Singapore, 2007: pp. 4138–4145. https://doi.org/10.1109/CEC.2007.4425011.

[9] S.C. Su, C.J. Lin, C.K. Ting, An effective hybrid of hill climbing and genetic algorithm for 2D triangular protein structure prediction, Proteome Sci. 9 (2011), S19. https://doi.org/10.1186/1477-5956-9-s1-s19.

[10] C. Huang, X. Yang, Z. He, Protein folding simulations of 2D HP model by the genetic algorithm based on optimal secondary structures, Comput. Biol. Chem. 34 (2010), 137–142. https://doi.org/10.1016/j.compbiolchem.2010.04.002.

[11] P.H.R. Gabriel, A.C.B. Delbem, Representations for evolutionary algorithms applied to protein structure prediction problem using HP model, in: K.S. Guimarães, A. Panchenko, T.M. Przytycka (Eds.), Advances in Bioinformatics and Computational Biology, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009: pp. 97–108. https://doi.org/10.1007/978-3-642-03223-3_9.

[12] T. Thalheim, D. Merkle, M. Middendorf, Protein folding in the hp-model solved with a hybrid population based aco algorithm, IAENG Int. J. Computer Sci. 35 (2008), 291–300.

[13] X. Zhang, W. Cheng, An improved Tabu search algorithm for 3D protein folding problem, in: T.B. Ho, Z.H. Zhou (Eds.), PRICAI 2008: Trends in Artificial Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008: pp. 1104–1109. https://doi.org/10.1007/978-3-540-89197-0_114.

[14] C. Rego, H. Li, F. Glover, A filter-and-fan approach to the 2D HP model of the protein folding problem, Ann. Oper. Res. 188 (2011), 389–414. https://doi.org/10.1007/s10479-009-0666-5.

[15] L.F. Nunes, L.C. Galvao, H.S. Lopes, et al. An integer programming model for protein structure prediction using the 3D-HP side chain model, Discr. Appl. Math. 198 (2016), 206–214. https://doi.org/10.1016/j.dam.2015.06.021.

[16] Y. Guo, F. Tao, Z. Wu, et al. Hybrid method to solve HP model on 3D lattice and to probe protein stability upon amino acid mutations, BMC Syst. Biol. 11 (2017), 93. https://doi.org/10.1186/s12918-017-0459-4.

[17] A.E. Smith, D.W. Coit, Penalty functions, Handbook on Evolutionary Computation, Section C 5.2, Oxford University Press and Institute of Physics Publishing, 1997.

[18] F. Glover, Future paths for integer programming and links to artificial intelligence, Computers Oper. Res. 13 (1986), 533–549. https://doi.org/10.1016/0305-0548(86)90048-1.

[19] F. Glover, Tabu search–Part I, ORSA J. Comput. 1 (1989), 190–206. https://doi.org/10.1287/ijoc.1.3.190.

[20] F. Glover, Tabu search–Part II, ORSA J. Comput. 2 (1990), 4–32. https://doi.org/10.1287/ijoc.2.1.4.

[21] C. Zhou, C. Hou, Q. Zhang, et al. Enhanced hybrid search algorithm for protein structure prediction using the 3D-HP lattice model, J. Mol. Model. 19 (2013), 3883–3891. https://doi.org/10.1007/s00894-013-1907-8.