ALGORITHMS FOR 3DHP PROTEIN STRUCTURE PREDICTION

**Luiz Fernando Nunes[*, 1, 2], Lauro Cesar Galvão[1, 2], Heitor Silvério Lopes[1],
Pablo Moscato[2, 3, 4], Regina Berretta[2, 3]**

[1]Bioinformatics Laboratory, Federal University of Technology – Paraná, Curitiba, Brazil.
[2]Centre for Bioinformatics, Biomarker Discovery and Information-based Medicine, The University of Newcastle, Australia.
[3]Hunter Medical Research Institute.
[4]ARC Centre of Excellence in Bioinformatics.
[*]e-mail: nunes@utfpr.edu.br

ABSTRACT

Proteins are fundamental compounds for all living beings. To predict the structure of a protein from its sequence of amino acids is a difficult task and time consuming. Therefore, computational approaches have been used. In recent years many simplified models were proposed to represent proteins in a computationally feasible way. In this work, we propose a greedy algorithm for 3DHP protein structure prediction problem. We have tested 10 instances with 27 amino acids each, and the computational time did not exceed 1 second in any case. To evaluate the results, we used two methods presented in the recent literature. In addition, optimal results for a fixed lattice were obtained using an integer linear programming model running with a comercial software. Although quite simple, the proposed greedy method obtained the same results when compared with the above mentioned optimal results, but requiring a much smaller computational time.

KEYWORDS. Greedy algorithm, protein structure prediction, integer linear programming.

1. Introduction

The proteins are present in all living systems and are composed of amino acid residues. There are 20 differents amino acids that constitute the building blocks of proteins and it is known that the function of a protein depends on the way that it spatially configures in a three-dimensional structure. One of the greatest challenges in Biology, Medicine and Biochemistry is to understand the process of how proteins fold. Nowadays it is not possible to determine the three-dimensional structure using the full analytic atomic model in the most general case.

Whereas proteins have a very complex structure, several models have emerged in recent years to simplify its representation. Researchers have developed several discrete models to find the optimal or quasi-optimal solutions for the Protein Structure Prediction Problem and to have a better understanding of the computational complexity of the core problem [Chandru *et al*. 2003], [Dinner *et al*. 2000]. In fact, it has been shown that even the simplest model proposed for representing proteins – bi-dimensional Hydrophobic-Polar (2DHP) model – is NP-complete.

Lattice models are the simplest models for the Protein Structure Prediction Problem, representing a protein as a sequence of chained elements each one of which is in a grid point on a lattice that can be in the plane (2D) or in the space (3D). Each position in the lattice can be occupied, at most, by one amino acid (which in the model are of only two types, hydrophobic or hydrophilic) and all successive pairs of amino acids on the chain must be positioning in grid points that are nearest neighbors of each other in the lattice.

The objective of a Prediction Structure Protein Problem is to minimize the free-energy of a conformation. The number of non-local bonds [Dill *et al*. 1995] or H-H contacts is inversely proportional to the free energy. Therefore, the objective is to maximize these interactions between non-adjacent hydrophobic amino acids of the sequence.

Although square lattices and cubic lattices are the most studied types of lattices, there are another models that use other type of lattices, such as triangular [Li *et al*. 2005], [Santos and Santos 2004] and hexagonal [Jiang and Zhu 2005]. Even with significant simplifications of lattice models, the 2DHP model and 3DHP model have some behavioral equivalency with real-world proteins [Dinner *et al*. 2000], [Dill *et al*. 1995], [Dobson and Karplus 1999]. There are also studies in the literature involving methods to solve problems considering side chains [Nunes *et al*. 2016].

2. The 2DHP and 3DHP Model

The 2DHP e 3DHP models were introduced in 1989 by Lau and Dill, 1989. The 2DHP [Crescenzi *et al*. 1998], [Nayak *et al*. 1998], [Ngo *et al*. 1994], [Unger and Moult 1993] and 3DHP [Atkins and Hart 1999], [Berger and Leighton 1998] models are the most studied discrete models for the protein structure prediction. In most cases, the Hydrophobic-Polar (HP) models use a square lattice or a cubic lattice where the amino acids can be assigned to grid points in the lattice. Even being very simple, the protein structure prediction problem using these models are NP-hard and many heuristics and metaheuristics approaches have been developed to tackle them [Chandru *et al*. 2003], [Dill *et al*. 1995], [Bitello and Lopes 2006], [Lopes and Scapin 2005]), [Lyngs and Pedersen 1999], [Tang 2000]. A common feature of optimization strategies for the 2DHP and 3DHP models is to obtain reductions on the free energy by employing strategies that keep hydrophobic amino acids in the inner of the protein, "protected" by surrounding hydrophilic amino acids.

As we said before, there are two types of amino acids in this model: hydrophobic (H) or polar (P). The polar amino acids are hydrophilic. The primary structure of a protein is usually represented by a string formed by elements belonged to the set {H, P}. We note that in the literature other variants have been studied of this basic model by including different hydrophobicity scales or extended alphabets to represent the physical and chemical properties of the amino acids [Backofen *et al*. 1999]. Figure 1 presents a conformation of a protein with 18 amino acids using the 3DHP model. Red and blue dots represent, respectively, the hydrophobic and hydrophilic amino acids. The chain is connected by black lines, and the bonds are represented by yellow lines. For this conformation there are 10 H-H contacts.

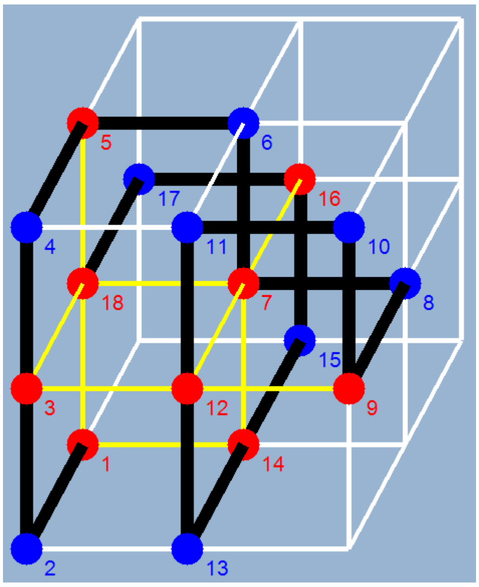


Fig. 1 - A valid 3D conformation of the protein chain defined by the HPHPHPHPHPPHPHPHPH.

3. The Proposed Method

In this work we present a new greedy algorithm to find the best possible conformation for a protein represented by the 3DHP model.

Pivot Moves

Pivot Moves are movements applied in a sequence of amino acids. From the set of amino acids, we chose one element that will be called the pivot. The pivot moves will be applied to other elements using the pivot element as reference. These movements can be a 90^o clockwise rotation, a counter-clockwise rotation, or 180^o. Each of these movements are defined by having as reference the axis parallel to the *x*, *y* or *z* axis that passes through the pivot. Fig. 2 illustrates the four possible pivot moves for a amino acid positioned in a cubic lattice. The pivot is highlighted in green in Fig 1-a. Figs. 2-b, 2-c and 2-d show a 90^o clockwise, counter clockwise rotations and a 180^o rotation, respectively. In this example, these three moves occurred in a horizontal plane. Fig. 2-e presents the last possible move for the pivot element 5.

Steps of the algorithm

Consider *n* the number of the amino acids, which can be hydrophobic or hydrophilic, and I and I_{max} the iteration and maximum number of iterations, respectively. This is an adaptation of the greedy algorithm presented in Galvão *et al*., 2012. However, in this work we are not considering the side chains. In that previous algorithm, there are two phases that are executed alternately. In the first phase a folding in the backbone sequence is done (without side chains). In the second phase is performed the position of the side chains. Now, we are going to use only the phase one of the previous method.

The following algorithm describes the steps of the method to solve the protein structure prediction problem in 3D (without considering side chains):

Step 1: Stretch: Set I = 0. At this stage the protein is fully stretched, such that the amino acids matches one of the axis of the coordinates (for instance assume it is the y-axis), i.e., each amino acid i will have coordinates $(0, i-1, 0), 1 \leq i \leq n$;

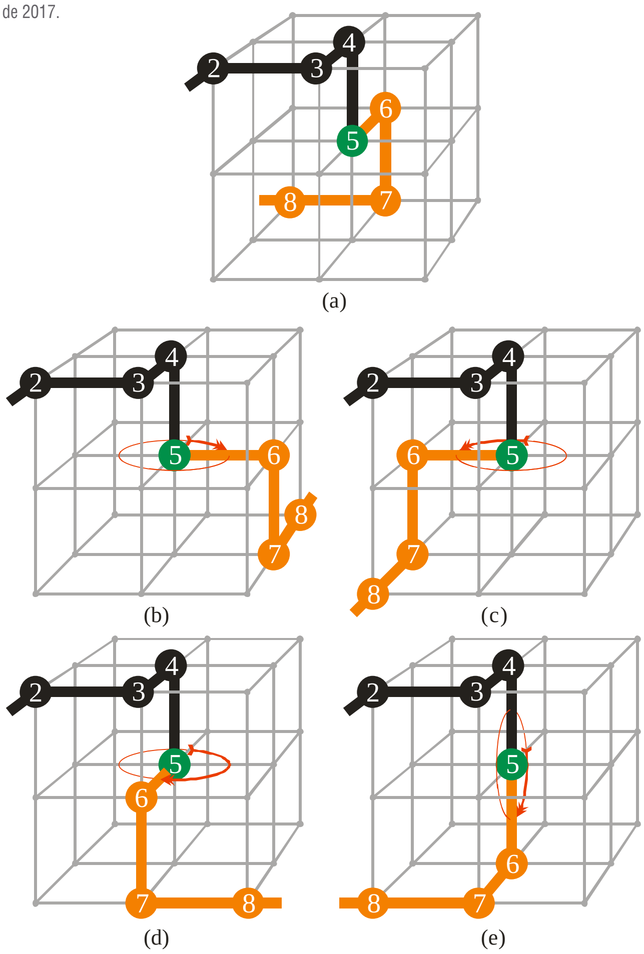
Step 2: Calculate E = the number of hydrophobic amino acids interactions;

Step 3: Select pivot: Set $I = I$+1. Choose a random point folding i, $2 \leq i \leq n-1$;

Step 4: Random move: Select uniformly at random one of the four feasible pivot moves that have not yet been tested in this iteration. If there are no untested pivot moves, go back to step 3;

Step 5: Test change: According to the move selected in Step 4, test the resulted 3D configuration as follows. For each amino acid element i+1 to n, check if there is any collision between the amino acids folded (i+1 to n) and the previous amino acids (1 to i). If there is a collision, reject the folding and go back to step 3; otherwise declare the pivot move as *'protein collision free'* and go to Step 6.

Step 6: Accept non-decreasing folding changes: Calculate $E'=$ the new number of interactions between all hydrophobic amino acids. If $E' \geq E$, the proposed folding is accepted and the largest number of hydrophobic interactions obtained is updated, i.e. $E = E'$. If $I = I_{\max}$, stop. Otherwise, go to *Step 3*.We run the algorithm (Step 1 to Step 6) Q times. The best value E obtained in these "Q rounds" of the algorithm is the lower bound for the maximum number of hydrophobic interactions for the particular instance under consideration.



(a)

(b)

(c)

(d)

(e)

Fig. 2 - Illustrations of some possible rotational movement from a pivot element.

4. An Integer Programming Model

Notation

Let *S* be a string with *n* elements each one belonging to the set $\{0,\ 1\}$. If *S* represents a protein then each element of the string indicates the hydrophobicity status of the amino acid (i.e. each hydrophobic amino acid is represented by the number one, while the number zero represents a hydrophilic amino acid).

We need to assign these amino acids to grid points of a 2D or 3D square lattice where each node can receive a maximum of one element. The finite nodes of this lattice are numerated from 1 to *m*, where *m* must be appropriately chosen to bear all the amino acids.

Let $I = \{1,\ \cdots, n\}$ be the set of indices in *S*. We partition $I = I_e \cup I_o$ such that I_e is the set of even indices and I_o the set of odd indices in *I*. Let H_e denote the set of indices of hydrophobic amino acids in the even positions in *I* and let H_o denote the set of indices of hydrophobic amino acids in the odd positions in *I*. Let $L = \{1,\ \cdots, m\}$ be the set of indices in the lattice. We partition $L = L_e \cup L_o$ such that L_e is the set of even elements and L_o is the set of odd elements in the *L*.

Let $N(v)$ the set of adjacent grid points to *v* in the lattice (neighborhood to *v*). So, $N(v) = \{\ t \in L\ /\ d(v, t) = 1\ \}$, where $d(v, t)$ is the Euclidean Distance between grid points *v* and *t*.

The nodes of the lattice are enumerated such that the neighborhood of an odd node in the lattice is formed only by even nodes and the neighborhood of an even node in the lattice is formed only by odd nodes.

For example, considering the lattice shown in the Fig. 3, the neighborhood of the vertex 14 is the set formed by the vertices 11, 17, 13, 15, 5 and 23. Likewise, the neighborhood of the vertex 2 is the set formed by the vertices 1, 3, 5 and 11. The set of feasible edges in the lattice is denoted by *E*, which is the set of (v, w) such that $v \in L_o$ and $w \in L_e, w \in N(v)$.

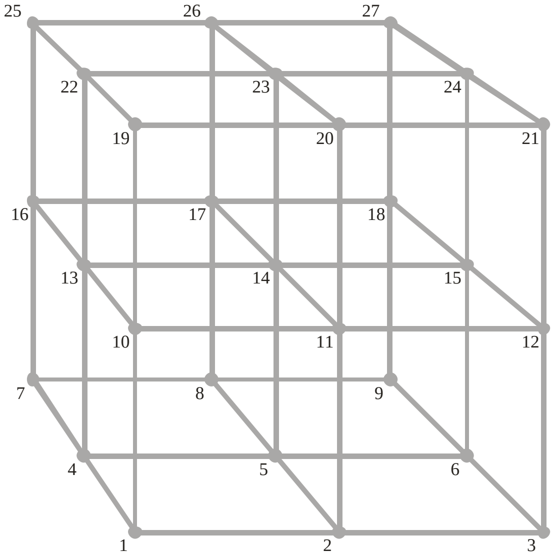


Fig. 3 - The nodes of the lattice are numerated so that the neighbourhood of an odd node in the lattice is formed only by even nodes and vice versa.

Variables for the Model

Without loss of generality, we consider that even amino acids are placed only on even lattice vertices, while odd amino acids are only placed on odd lattice nodes.

Let x_{iv} be a zero-one variable which is both defined for the case that either $i \in I_o$ and $v \in L_o$, or to $i \in I_e$ and $v \in L_e$; x_{iv} indicates whether or not the amino acid *i* is placed on the lattice node *v* (as a convention, x_{iv} is set 1 if the amino acid *i* is placed on lattice point *v* and 0 otherwise).

The variables $hh_{(iv)(jw)}$ are defined to $i \in H_o, j \in H_e - \{i-1, i+1\}$, with $(v, w) \in E$, indicating whether or not there is a contact between hydrophobic amino acids *i* and *j* on edge (v, w). So, $hh_{(iv)(jw)}$ is set to 1 if the there is a contact between hydrophobic amino acid *i* and *j* on edge (v, w) and 0 otherwise.

An Integer Formulation Program

The model (1)-(10) used to evaluate the results of the Greedy Algorithm is very similar to the models presented in [Carr and Hart 2002] and [Yanev *et al*. 2011].

$$\text{Max} \quad f = \sum_{(v,w)\in E} \sum_{i\in H_o} \sum_{j\in H_e-\{i-1,i+1\}} hh_{(iv)(jw)}$$

Subject to:

$$\sum_{v \in L_o} x_{iv} = 1 \qquad \forall i \in I_o \tag{2}$$

$$\sum_{v \in L_e} x_{iv} = 1 \qquad \forall i \in I_e \tag{3}$$

$$\sum_{i \in I_o} x_{iv} \leq 1 \qquad \forall v \in L_o \tag{4}$$

$$\sum_{i \in I_e} x_{iv} \leq 1 \qquad \forall v \in L_e \tag{5}$$

$$\sum_{w \in N(v)} x_{(i+1)w} \geq x_{iv} \qquad \forall i \in I_o - \{n\},\ v \in L_o \tag{6}$$

$$\sum_{w \in N(v)} x_{(i+1)w} \geq x_{iv} \qquad \forall i \in I_e - \{n\},\ v \in L_e \tag{7}$$

$$\sum_{j \in H_e} hh_{(iv)(jw)} \leq x_{iv} \qquad \forall i \in H_o, (v,w) \in E \tag{8}$$

$$\sum_{i \in H_o} hh_{(iv)(jw)} \leq x_{jw} \qquad \forall j \in H_e, (v,w) \in E \tag{9}$$

$$x_{iv},\ hh_{(iv)(jw)} \in \{0,1\} \tag{10}$$

The value of the objective function (1) represents the number of the non-local hydrophobic interactions. Constraints (2) and (3) guarantee that each amino acid *i* is assigned to exactly one node *v* in the lattice. Constraints (4) and (5) guarantee that each node *v* in the lattice contains at most one amino acid. Constraints (6) and (7) are used to force each amino acid consecutive on the string to be placed on an adjacent lattice point to its neighbour on the string. Constraints (8) to (9) are used to force elements to be placed on lattice nodes *v* and *w* if there is a contact between these elements on edge (v,w). Constraints (10) enforce that all the variables are integer.

5. Computacional Results

In Table 1 we present the benchmark sequences that have been used to evaluate our greedy heuristic algorithm for 3DHP protein structure prediction.

In Table 2 we present the computational results obtained. In the column “GH” are the results obtained with the greedy heuristic algorithm proposed in this work, using Visual Basic Language 11 (VB.Net) and a PC laptop running Windows 7, 64-bit and 6 GB of RAM with an Intel ® Core (TM) i7-2620M processor at 2.70GHz.

For all instances in the proposed algorithm was used Q = 40 and $I_{\max} = 5000$.

To evaluate these results, the optimal values were obtained using the integer formulation model previously presented. We have used ILOG CPLEX optimization package, version 12.4 to solve the integer programming problem in a high performance computing cluster with Dual Xeon 5550 2.67 GHz.

It is important to note that the optimal solutions obtained using ILOG CPLEX refer to a predetermined lattice (in this work, we have used lattices 5×5×5).

TABLE 1 - Benchmark instances for the 3DH

ID	*N*	Protein Sequence
Unger273d.1	27	$(PH)_3H_2P_2(HP)_2P_{10}H_2P$
Unger273d.2	27	$PH_2P_{10}H_2P_2H_2P_2HP_2HPH$
Unger273d.3	27	$H_4P_5HP_5H_3P_8H$
Unger273d.4	27	$H_3P_2H_4P_3(HP)_2PH_2P_2HP_3H_2$
Unger273d.5	27	$H_4P_4HPH_2P_3H_2P_{10}$
Unger273d.6	27	$HP_6HPH_3P_2H_2P_3HP_4HPH$
Unger273d.7	27	$HP_2HPH_2P_3HP_5HPH_2(PH)_3H$
Unger273d.8	27	$HP_{11}(HP)_2P_7HPH_2$
Unger273d.9	27	$P_7H_3P_3HPH_2P_3HP_2HP_3$
Unger273d.10	27	$P_5H(HP)_5(PHH)_2PHP_3$

The results obtained were also compared with the results obtained in [Guo and Feng 2006] and [Liu *et al*. 2012].

The results obtained by the proposed greedy algorithm were the same to the ones presented in Liu *et al*., 2012 and they were obtained in less than 1 second in all cases. In one case, our result was better than the result obtained in Guo and Feng, 2006.

In 9 cases, the software ILOG CPLEX also achieved the same results, showing the quality of our results. In only one case ILOG CPLEX was not able to find the optimal solution, and it was stopped when the dual gap was 41%, after 5 days of program execution.

TABLE 2 - Comparison of results

ID	*N*	EN[a]	Time EN[a] (sec)	HELP[b]	Time HELP[b] (sec)	CPLEX[c]	Time CPLEX[c]	Dimensions Lattice[c]	GH[d]	Time GH[d] (sec)
Unger273d.1	27	9	-	9	4.28	9	1,505	5×5×5	9	< 1
Unger273d.2	27	10	-	10	3.78	10	3,235	5×5×5	10	< 1
Unger273d.3	27	8	-	8	< 1	8	2,127	5×5×5	8	< 1
Unger273d.4	27	15	-	15	< 1	15	410,305	5×5×5	15	< 1
Unger273d.5	27	8	-	8	1.36	8	5,103	5×5×5	8	< 1
Unger273d.6	27	11	-	12	1.09	12	19,400	5×5×5	12	< 1
Unger273d.7	27	13	-	13	1.00	(>41%)	(*)	5×5×5	13	< 1
Unger273d.8	27	4	-	4	< 1	4	56	5×5×5	4	< 1
Unger273d.9	27	7	-	7	< 1	7	635	5×5×5	7	< 1
Unger273d.10	27	11	-	11	< 1	11	5,919	5×5×5	11	< 1

[a] Values are from [Guo and Feng 2006], [b] Values are from [Liu *et al*. 2012], [c] Values are from the present work using CPLEX , [d] Values are from the present work using GH.

6. Conclusions

The results obtained using the ILOG CPLEX were optimal for the lattices used. The use of larger lattices could eventually provide a better result. In general, however, this is unlikely, as there is a tendency (in the used model) of hydrophobic amino acids being positioned internally in the protein, "protected" from water by the hydrophilic amino acids.

These results show the quality of the method presented in this work, that were the same results obtained in [Liu *et al*. 2012]), but with computational time inferior in half of the instances. If we compare the results with the method shown in [Guo and Feng 2006], we have a better result in one case (see Fig. 4) and the same results in the remaining cases. The computational times are not reported in [Guo and Feng 2006]. Our proposed method is an alternative to the previous ones. We believe this process can be improved with the use of local algorithm improvements. The use of CPLEX can facilitate a proof of optimality for some instances. For larger instances there is still room to improve in terms of speeding-up these exact algorithms.

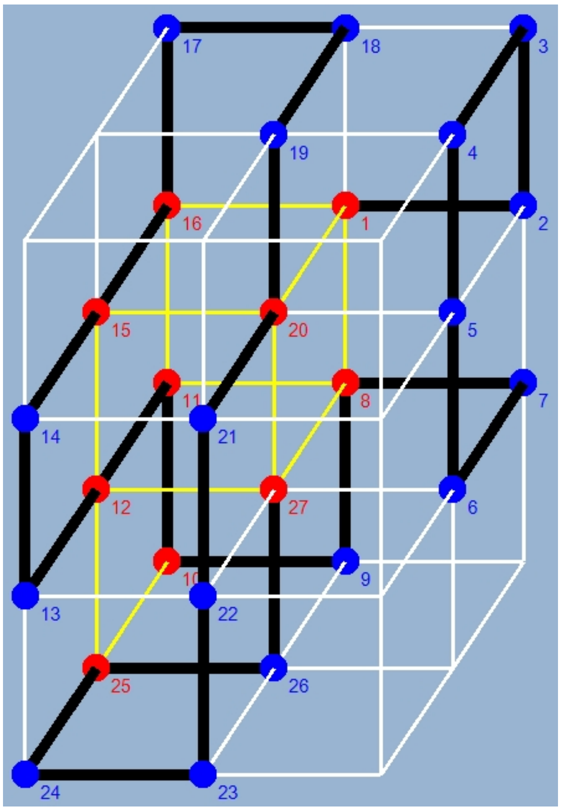


Fig. 4 - The result obtained to the sequence Unger273d.6. It can be observed that the hydrophobic amino acids were concentrated in the central part of the molecule, while the hydrophilic amino acids were in the outer part. The non-local hydrophobic interactions are indicated by yellow lines.

7. Acknowledgment

Authors would like to thanks the Brazilian National Research Council – CNPq, for the financial support under grants scholarship 202312/2011-9 to L. F. Nunes and the ARC Centre of Excellence in Bioinformatics to P. Moscato and the Federal University of Technology – Paraná – UTFPR to L. C. Galvão and L. F. Nunes.

References

Atkins, J. and Hart, W.E. (1999). On the Intractability of Protein Folding with a Finite Alphabet of Amino Acids. *Algorithmica,* 25:279-294.

Backofen, R. *et al*. (1999). Application of constraint programming tecniques for structure prediction of lattice proteins with extended alphabets. *Bioinformatics*, 15:234-242.

Berger, B. and Leighton, F. T. (1998). Protein Folding in the Hydrophobic-Hydrophilic (*HP*) Model is NP-Complete. *Journal of Computational Biology*, 5:27-40.

Bitello, R. & Lopes, H. S. (2006). A diferential evolution approach for protein folding. In. *Proc. IEEE Symp. on Computational Intelligence in Bioinformatics and Computational Biology*, p. 1-5.

Carr, B. & Hart, B. (2002). Discrete Optimization Models for Protein Folding. *Sandia Report*, Alantha Newman, MIT, August.

Chandru, V. *et al*. (2003). The algorithms of folding proteins on lattices. *Discrete Applied Mathematics Math*., 127:145-161.

Crescenzi, P. *et al*. (1998). On the Complexity of Protein Folding. *Journal of Computational Biology*, 5:423-465.

Dill, K. A. *et al*. (1995). Principles of protein folding – A perspective from simple exact models. *Protein Science*, 4:561-602.

Dinner, A. R. *et al*. (2000). Understanding protein folding via free-energy surfaces from theory and experiment. *Trends in Biochemical Sciences*, 25:331-339.

Dobson, C. M. and Karplus, M. (1999). The fundamentals of protein folding: bringing together theory and experiment. *Current Opinion in Structural Biology*, 9:92-101.

Galvão, L. C. *et al*. (2012). A New Greedy Heuristic for 3DHP Protein Structure Prediction With Side Chain. In: *Proceedings of The 2012 Computational Structural Bioinformatics Workshop in conjunction with the IEEE International Conference on Bioinformatics and Biomedicine*, Philadelphia, PA, USA.

Guo, Y. Z. and Feng, E. M. (2006). The simulation of the three-dimensional lattice hydrophobic-polar protein folding. *The Journal of Chemical Physics*, 125.

Jiang, M. and B. Zhu, B. (2005). Protein Folding on the Hexagonal Latice in the HP Model. *Journal of Bioinformatics and Computational Biology*, 3:19-34.

Lau, K. and Dill, K. (1989). A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22:3986-3997.

Li, Z. *et al*. (2005). Unique Optimal Foldings of Proteins on a Triangular Lattice. *Applied Bioinformatics*, 4:105-116.

Liu, J. *et al*. (2012). Heuristic energy landscape paving for protein folding problem in the three-dimensional HP lattice model. *Computational Biology and Chemistry*, 38:17-26.

Lopes, H. S. and Scapin, M. P. (2005). An Enhanced Genetic Algorithm for protein Structure Prediction Using the 2D Hydrophobic-Polar Model. *Lecture Notes in Computer Science.* 3871: 238-246.

Lyngs, R. B. and Pedersen, C. N. S. (1999). Protein folding in the 2D HP model. *Technical Report RS-99-16, BRICS Bioinformatics Research Center*, University of Aarhus.

Nayak, A. *et al*. (1998). Spatial codes and the hardness of string folding problems. In: *Proc. 9th Ann. Symp. on Discrete Algorithms*, p. 639-648.

Ngo, J. T. *et al*. (1994).Computational complexity, protein structure prediction, and the Levinthal paradox. In: *The Protein folding problem and terciary structure prediction*, Ed. K. Merz Junior and S. LeGrand, Birkhuser, Boston.

Nunes, L. F. *et al*. (2016). An integer programming model for protein structure prediction using the 3D-HP side chain model. *Discrete Applied Mathematics*^{JCR}, 198:206-214.

Santos, E. E. and Santos Jr. E. (2004). Reducing the computational load of energy evaluations for protein folding. In: *Proc. 4th Symp. on Bioinformatics and Bioingineering*, p. 79-86.

Tang, C. (2000). Simple models of the protein folding problem. *Physica A: Statistical Mechanics and its applications*, 288:31-48.

Unger, R. & Moult, J. (1993). Finding the lowest free energy conformation of a protein is a NP-hard problem: proof and implications. *Bulletin of Mathematical Biology*, 55:1183-1198.

Yanev, N. *et al*. (2011). Integer programming approaches to HP folding. In: *VIII European Workshop in Drug Design, Certosa di Pontignano – Siena, May 22nd-28th*.