단백질 접힘 현상을 예측하기 위한 Hydrophobic-Hydrophilic 모델에서 수리모델을 활용하여 목적함수 값의 상한 값 구하기

An Upper Bound for the Number of Contacts in Hydrophobic-Hydrophilic Lattice Protein Structure Prediction Model Using a New Mathematical Formulation

안남수, 박성수
한국과학기술원 산업 및 시스템 공학과
대전 광역시 유성구 과학로 335 산업 및 시스템 공학과, 305-701

Abstract

Protein is a complex biological macromolecule that is composed of a sequence of amino acids. Proteins perform many important roles in all living organism such as structural components, active agents, and enzymes. Since it is believed that the functional properties of the protein are depending on its structure, it is critical to predict the protein's structure to understand the functional properties.
One of the most widely studied protein structure prediction models is the hydrophobic-hydrophilic (HP) model, which abstracts a dominant force in protein folding. That is, to explain the hydrophobic interaction, HP model tries to maximize contacts among hydrophobic amino acids. Unfortunately, finding an optimal folding with HP model is shown to be NP-complete for both in 2D square and 3D cubic lattices. Although huge number of previous researches tried to find a tight lower bound for the number of contacts using heuristic methods, these methods cannot guarantee the quality of the obtained solution since no information on upper bound has been obtained.
In this research, we focus on identifying the efficiently computable upper bound. We present a new mathematical formulation for HP model, which can provide an upper bound using linear relaxation of the formulation. Compared with other existing mathematical formulation, computational experiments show that our formulation provides much tight upper bound. In particular, we discuss how the linear relaxation upper bound can be improved, and preliminary computational results using benchmark problems are reported.

1. Introduction

A protein is a complex biological macromolecule composed of a sequence of amino acids. Proteins play key functions in many living organisms. For example, proteins can act as either structural component such as hair, skin and muscles, or play a role of active agents such as enzymes and transporting oxygen to our tissues.

Amino acids are joined end-to-end, and the sequence forms a backbone of the protein. The sequence of acids in a protein is called its primary structure. Under certain standard conditions, proteins always fold to the same unique native three-dimensional (3D) structure, and surprisingly, this shape is principally determined by the sequence of amino acids. This was shown by Christian Anfinsen, who won the 1972 Nobel Prize in Chemistry for this work. For that reason, it is believed that the functional properties of a protein depend on its 3D structure. However, experimental methods frequently used to determine the 3D structure of a protein such as X-ray crystallography and nuclear magnetic resonance spectroscopy are very expensive and time consuming. Also, they can be performed under standard conditions and some important proteins are difficult or impossible to crystallize, so we do not know their 3D structure. This has lead to an enormous research interest in the development of methods to predict 3D structure of protein from the sequence information.

In summary, given the sequence information of amino acids, finding the native structure of the protein is called protein structure prediction (PSP) problem, and many models are suggested to address the PSP problem. Among suggested models, lattice models, which embed amino acids on a lattice, can be used to extract essential principles, make predictions and unify our understanding of many different properties of proteins (Dill et al., 1995).

In this research, we study hydrophobic-hydrophilic (HP) model on two-dimensional (2D) square lattice, which is one of the most extensively studied lattice models. HP model was first introduced by Dill (1985). HP model classifies the 20 amino acids as hydrophobic (H) and hydrophilic (P), and this classification is known from experimental results. Therefore, protein can be modeled as a sequence of H's and P's, and this sequence is embedded on the lattice without self-intersecting. A pair of amino acids that placed in successive positions in the sequence is called sequential neighboring, and a pair of non-successive amino acids that are adjacent in the embedding on the lattice is called topological neighboring (Chandru et al., 2004). HP model

maximizes the number of HH pairs of topological neighboring. That is, HP model abstracts the hydrophobic interaction, which is one of the principal forces in protein folding (Alberts et al., 1998). Despite its simplicity, HP model is known to be powerful enough to capture a variety of properties of actual proteins and has been used to discover new properties (Hart and Newman, 2001).

This paper is organized as follows. In Section 2, brief overview of previous researches on HP model will be presented. In Section 3, we present a new mathematical formulation for HP model. In Section 4, to improve the upper bound using the linear programming relaxation, we introduce additional constraints. Section 4 contains computational experiments and it show that our formulation provides tight upper bound. Also, in Section 4, we discuss how the upper bound using the linear programming relaxation can be improved. Finally, in Section 5, contribution of our research is summarized and possible direction for future research will be introduced.

2. Literature Review

Although protein folds in 3D space, scientists often work with 2D model instead of 3D model to verify the performance of their algorithm. For convenience, this review limits on HP model in 2D square lattice.

Since the HP model on 2D square lattice was shown to be NP-complete by Crescenzi et al. in 1998, most previous researches were focused on identifying a tight lower bound for the HH pairs of topological neighboring using heuristic methods. Almost meta-heuristic algorithms such as genetic algorithm by Bui and Sundarraj in 2005, simulated annealing by Li in 2007, tabu search by Lesh et al. in 2003, ant colony by Shmygelska et al. in 2003, and local search by Guo et al. in 2006 were applied to handle 2D HP square lattice model. Also, heuristic algorithms, so called a filter-and-fan approach by Rego et al. in 2006 and immune algorithm by Cutello et al. in 2007 were developed. Among various heuristic methods, a filter-and-fan approach performs best in benchmark problems without considering computing efficiency.

Few performance-guaranteed approximation algorithms can be found. Hart and Istrail (1996) developed 1/4 approximation ratio algorithm, and later, Newman (2002) improved this ratio to 1/3.

Although solving HP model exactly is an important research goal, not many exact approaches can be found. Yue and Dill (1993) developed first exact method, which enumerates all possible shapes of the given H amino acids. However, Backofen and Will (2006) showed that the method designed by Yue and Dill is an incomplete one. Yoon (2006) provided a model for constraint programming and developed some techniques to solve the model faster, but it cannot be used to solve large-scale instances exactly. Carr et al. (2003) and Greenberg et al. (2004) presented integer programming formulations for HP model. However, since their formulations require huge number of variables, even small sized instances cannot be addressed.

In this research, we present a mathematical formulation for HP model, which can be used to provide an upper bound using linear relaxation of the formulation. Compared with the mathematical formulation suggested in Yoon (2006), our formulation provides tight upper bound of 2 to 10 orders of magnitude with respect to benchmark problems.

3. Mathematical Formulation

This section presents a mathematical formulation for HP model. For simplicity, the model that we use is the HP model on 2D square lattice, but it can be easily extended to HP model on 3D cubic lattice. This extension technique will be covered briefly at the end of Section 3.2. We first introduce how the HP lattice model can be implemented using a mathematical formulation. Then, some additional constraints for the mathematical formulation are introduced.

3.1 Integer programming formulation

HP model for PSP problem maximizes the number of HH topological contacts while satisfying the following three restrictions. First, each amino acid must be assigned in one of lattice points. Second, each lattice point cannot be allocated more than one amino acid. Third, every two amino acids that are consecutive in the protein's sequence must be assigned to a neighboring lattice points.

Integer programming formulation suggested in this research uses a direction from a current amino acid to the next sequential neighboring acid. Since we start with 2D square lattice, we have only four directions: up, down, left and right.

The decision variables are defined as follows.

$$x_{i,i+1,d} = \begin{cases} 1, \text{if direction between } i^{\text{th}} \text{ amino acid and } (i+1)^{\text{th}} \\ \quad \text{amino acid is } d\,, \\ 0, \text{otherwise.} \end{cases}$$

Also, we need integer variables to indicate the L_1 distance of each pair of amino acids in 2D square lattice.

$$z_{i,j,d,k} = \begin{cases} 1, \text{if the } L_1 \text{ distance between } i^{\text{th}} \text{ amino acid and } (i+1)^{\text{th}} \\ \quad \text{amino acid is } k \text{ in } d \text{ direction,} \\ 0, \text{otherwise.} \end{cases}$$

To count the number of HH pairs of topological

contacts, we need additional integer variables.

$c_{i,j,d} =$

$$\begin{cases} 1, \text{if direction of contact between } i^{\text{th}} \text{ H amino acid and } j^{\text{th}} \text{ H amino acid is } d, \\ 0, \text{otherwise.} \end{cases}$$

Lastly, following notation is necessary to formulate the model compactly, and it will be used throughout the paper.

$H \stackrel{\text{def}}{=}$ set of increasing order of H acids' indices HP $\stackrel{\text{def}}{=}$ set of increasing order of all amino acids' indices

$$D \stackrel{\text{def}}{=} \{\text{up}, \text{down}, \text{right}, \text{left}\}$$

$$S \stackrel{\text{def}}{=} \{(i,j) | \, i \in HP, j \in HP, j \geq i+1\}$$

Since 2D square lattice is a bipartite graph, each odd index H amino acid can have contacts with only even index H amino acids. Similarly, each even index H amino acid can have contacts with only odd index H amino acids. Thus, set of possible topological contacts between any two H acids can be defined as

$$C \stackrel{\text{def}}{=} \{(i,j) | \, i \in H, j \in H, j \geq i+3, (j-i) = 2l+1 (l \geq 1)\}.$$

Now, an integer programming formulation is given as follows:

Maximize

$$\sum_{(i,j)\in C} \sum_{d \in D} c_{i,j,d}, \quad (1)$$

Subject to

$$\sum_{d \in D} x_{i,i+1,d} = 1, \quad \forall \text{i} = 1, \dots, |HP|-1. \quad (2)$$

$$\sum_{k=1}^{\text{j-i}} k z_{i,j,d,k} - \sum_{k=1}^{j-i} k z_{i,j,\bar{d},k} = \sum_{k=i}^{j-1} (x_{k,k+1,d} - x_{k,k+1,\bar{d}}), \quad \forall (i,j) \in S, d \in \{\text{up}, \text{right}\}. \quad (3)$$

$$\sum_{k=1}^{\text{j-i}} z_{i,j,d,k} + \sum_{k=1}^{j-i} z_{i,j,\bar{d},k} \leq 1, \quad \forall (i,j) \in S, d \in \{\text{up}, \text{right}\}. \quad (4)$$

$$\sum_{d \in D} \sum_{k=1}^{j-i} z_{i,j,d,k} \geq 1, \quad \forall (i,j) \in S. \quad (5)$$

$$\sum_{d \in D} c_{i,j,d} \leq 1 + \frac{1 - \sum_{d \in D} \sum_{k=1}^{j-i} k z_{i,j,d,k}}{M}, \quad \forall (i,j) \in C. \quad (6)$$

$$c_{i,j,d} \leq z_{i,j,d,1}, \quad \forall (i,j) \in C, d \in D. \quad (7)$$

$$x_{i,i+1,d} \in \{0,1\}, \quad \forall (i,j) \in C, d \in D. \quad (8)$$

$$z_{i,j,d,k} \in \{0,1\}, \quad \forall (i,j) \in S, d \in D, k = 1, \dots, j-i. \quad (9)$$

$$c_{i,j,d} \in \{0,1\}, \quad \forall (i,j) \in C, d \in D. \quad (10)$$

Constraint (2) defines that, only one direction can be chosen between two consecutive amino acids. Constraint (3) defines the $z_{i,j,d,k}$ variable, which is a L_1 distance in 2D square distance between the i^{th} amino acid and j^{th} amino acid in d direction. Note that $\bar{d}$ indicates the opposite direction of d. Therefore, both $z_{i,j,d,k}$ and $z_{i,j,\bar{d},k}$ cannot take one as a decision value at the same time, if constraint (4) holds. To guarantee the non-overlapping restriction between any pair of amino acids, constraint (5) is necessary. Constraint (6) and (7) define the $c_{i,j,d}$ variable. Constraint (6) indicates that, if the L_1 distance between i^{th} H amino acid and j^{th} H amino acid in 2D square lattice is exactly one, $\sum_{d \in D} c_{i,j,d}$ takes one(since we are maximizing $\sum_{(i,j)\in C} \sum_{d \in D} c_{i,j,d}$). However, if the L_1 distance between i^{th} H amino acid and j^{th} H amino acid is greater than one, $\sum_{d \in D} c_{i,j,d}$ takes zero. Constraint (7) enforces that, if $c_{i,j,d}$ takes one, corresponding $z_{i,j,d,1}$ takes one as a decision value. The objective function (1) maximizes the number of possible topological HH contacts.

This formulation for HP model on 2D square lattice can be easily extend to the 3D cubic lattice formulation by using six directions (up, down, right, left, forward, and backward) instead of using four directions.

3.2 Additional constraints

Before running the proposed integer programming formulation, following constraints can be added to strengthen the formulation.

Since 2D square lattice is a bipartite graph, if two amino acids have an even index difference in the HP sequence, they cannot be adjacent in the lattice. That is, their L_1 distance in the 2D square lattice should be greater than or equal to two. Therefore, following inequalities can be added.

$$\sum_{d \in D} \sum_{k=1}^{j-i} k z_{i,j,d,k} \geq 2, \quad \forall (i,j) \in S, (j-i) = 2l (l \geq 1) \quad (11)$$

Likewise, if the index difference between two amino acids is one in the HP sequence, L_1 distance in the 2D square lattice between two amino acids should be one. In the same manner, if the index difference between two amino acids is two in the sequence, L_1 distance in the 2D square lattice between two amino acids should be two. However, this relationship does not hold when the distance between two amino acids is greater than two in the sequence.

$$\sum_{d \in D} z_{i,i+1,d,1} = 1, \quad \forall i = 1, \dots, |HP|-1. \quad (12)$$

$$\sum_{d \in D} \sum_{k=1}^{2} k z_{i,i+2,d,k} = 2, \quad \forall i = 1, \dots, |HP|-2. \quad (13)$$

Note again that, we work on 2D square lattice. Therefore, each H amino acid that is not in the first or last position in the sequence can have at most two topological HH contacts. If H amino acid is located in the first or last position in the sequence, it can have at

most three topological HH contacts. To represent these relationships, we define other notations as follows.

$$F(i) \stackrel{\text{def}}{=} \{j \mid i \in H, j \geq i+3, (j-i) = 2l+1 (l \geq 1)\}.$$

$$B(i) \stackrel{\text{def}}{=} \{g \mid g \in H, g \leq i-3, (i-g) = 2l+1 (l \geq 1)\}.$$

Now the constraints are,

$$\sum_{j \in F(i)} \sum_{d \in D} c_{i,j,d} + \sum_{g \in B(i)} \sum_{d \in D} c_{g,i,d} \leq 2, \quad \forall\, i \in H, 2 \leq i \leq |HP| - 1. \tag{14}$$

$$\sum_{j \in F(1)} \sum_{d \in D} c_{1,j,d} \leq 3, \quad 1 \in H. \tag{15}$$

$$\sum_{g \in B(|HP|)} \sum_{d \in D} c_{g,|HP|,d} \leq 3, \quad |HP| \in \text{H}. \tag{16}$$

Also, for a given i^{th} amino acid and d direction, $x_{i,i+1,d}$, $x_{i-1,i,\bar{d}}$, $\sum_{j \in F(i)} c_{i,j,d}$, and $\sum_{g \in B(i)} c_{g,i,\bar{d}}$ cannot take a value of one at the same time because of non-overlapping requirement. Figure 1 illustrates this situation for a given i^{th} amino acid and *up* direction.

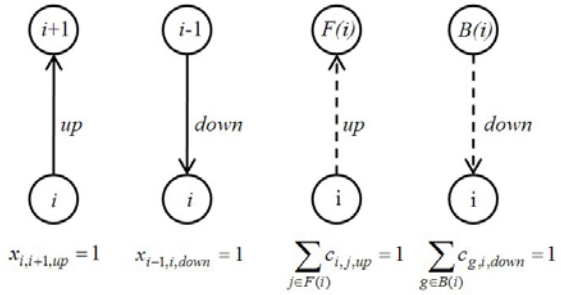

Figure 1. Non-overlapping requirement for a given i^{th} amino acid and *up* direction

Thus, we have

$$x_{i,i+1,d} + x_{i-1,i,\bar{d}} + \sum_{j \in F(i)} c_{i,j,d} + \sum_{g \in B(i)} c_{g,i,\bar{d}} \leq 1, \quad \forall\, i \in H, d \in D. \tag{17}$$

Lastly, below relationship can be established according to the definition of $x_{i,i+1,d}$ and $z_{i,j,d,k}$ variables.

$$x_{i,i+1,d} = z_{i,i+1,d,1}, \quad \forall i = 1, \ldots, |HP| - 1, d \in D. \tag{18}$$

4. Computational Results

In this section, we present an upper bound for various benchmark problems using linear relaxation of the mathematical formulation. Computational experiments show that our formulation provides tight upper bound. We also discuss how the linear programming relaxation upper bound can be improved by enforcing integrality restriction on some variables instead of all variables used in the mathematical formulation.

Benchmark problems were taken from http://www.cs.sandia.gov/tech_reports/compbio/tortilla-hp-benchmarks.html.

In Table 1, First column indicates the instance that we used, and second column shows that the best lower bound from heuristic algorithm suggested by Rego et al (2006). Third and fourth columns denotes the upper bounds using linear programming relaxation of the mathematical formulation suggested in this research without additional constraints and with additional constraints, individually.

Fifth column means the gap between the lower bound and upper bound with additional constraints. In some problems, we cannot find any heuristic algorithms, which are applied to solve the instances. Those cases are denoted by N/A sign.

All experiments were run on AMD Athlon™ 64 X2 Dual Core (2.70GHz) with 2GB Ram, and we used Xpress-MP2007a as optimization software.

Table 1. Computational results for benchmark problems

Problem ($\|HP\|, \|H\|$)	Lower bound	Upper bound without additional constraints	Upper bound with additional constraints	Gap
20, 10	9	23	11	2
24, 10	9	23	11	2
25, 9	8	16	8	0
36, 16	14	54	16	2
48, 25	23	139	25	2
50, 24	21	136	25	4
60, 43	36	425	40	4
64, 42	42	413	43	1
102, 37	N/A	325	36	N/A
123, 47	N/A	521	36	N/A
136, 50	N/A	602	49	N/A

From Table 1, we can find that the additional constraints improve the objective function value significantly, and the gap between the lower bound and upper bound with additional constraints is very small. Small gap between the lower bound and upper bound proves that the linear relaxation of our formulation provides very tight upper bound. Since information on upper bound is provided, existing heuristic algorithms can guarantee the quality of the obtained solutions. However, when the HP sequence consists of H amino acid only, our formulation does not generate tight upper bound any more. We arbitrarily generate the problem, and obtained upper bound using the linear programming relaxation as shown in Table 2. Optimal value is obtained by running the optimization software.

Table 2. Computational result for hard problem

Problem ($\|HP\|, \|H\|$)	Upper bound	Optimal
12, 12	13	6

Table 2 shows that the gap between the upper bound and optimal is not small any more. To resolve this difficulty, we suggest adding integrality condition on some variables instead of all variables. Following Table 3 supports our idea, and shows how it works. Each solution is obtained by using the optimization software.

Table 3. Computational result of adding integrality condition

Problem($\|HP\|, \|H\|$)	12, 12
Upper bound from linear relaxation	13
Optimal solution (objective function value, nodes, CPU time)	6, 98473 nodes, 1013.1 second
Integer $x_{i,i+1,d}$ variables (objective function value, nodes, CPU time)	13, 23 nodes, 1.4 second
Integer $z_{i,j,d,k}$ variables (objective function value, nodes, CPU time)	6, 1668753 nodes, 9429.5 second
Integer $c_{i,j,d}$ variables (objective function value, nodes, CPU time)	6, 34787 nodes, 267.3 second

From Table 3, we can find that, integral restriction on $c_{i,j,d}$ or $z_{i,j,d,k}$variables generate very tight upper bound. In fact, that upper bound is the same value as the optimal value of the objective function. Note that, when we put the integral restriction on $c_{i,j,d}$ variable, the CPU time was much smaller than the CPU time of the optimal solution or CPU time of the integral restriction on $z_{i,j,d,k}$ variable.

Therefore, to strengthen linear relaxation upper bound, solving the problem with integral restriction on $c_{i,j,d}$ variables seems effective. We applied the same idea on one of benchmark problems.

Table 4. Computational result of adding integrality condition for one of benchmark problems

Problem($\|HP\|, \|H\|$)	20, 10
Lower bound	9
Upper bound from linear relaxation	11
Integer $c_{i,j,d}$ variables (objective function value, nodes, CPU time)	9, 57387 nodes, 3058 second
Optimal solution (objective function value, nodes, CPU time)	8, 360832 nodes, >36000 second

Since the improved upper bound is 9, which is the same as the lower bound from the heuristic algorithm, we can conclude that the generated solution from the heuristic algorithm is an optimal solution.

However, adding the integral restriction on $c_{i,j,d}$ variable and solve it to optimality is a mixed integer linear programming problem (MILP), and may hard to solve it optimally. Therefore, further research on this MILP is necessary.

5. Conclusion and Future Works

In this research, we suggested a mathematical formulation for HP lattice model of PSP problem, which can be used in 2D square lattice. Also, our model can be easily extended to 3D cubic lattice by modifying some of the constraints. Our mathematical formulation can provide an upper bound for HP model using linear programming relaxation. Compared with other existing mathematical formulation, computational experiments on benchmark problems show that our formulation provides much tight upper bound.

Upper bound from the linear relaxation of our model can provide information about the quality of the obtained solution from the existing heuristic algorithms. Furthermore, if the upper bound is the same as lower bound, we can conclude that the obtained solution from the heuristic algorithm is an optimal solution. Therefore, improving the upper bound is an important research topic which needs to be performed as a future research. To address this issue, we suggested to putting integral restriction on small part of variables and preliminary computational result is reported.

References

A. Shmygelska, R.A. Hernandez, H.H. Hoos. (2002), *An ant colony optimization algorithm for the 2D HP protein folding problem*, LNCS 2463, 40-52.

Alantha Newman. (2002), *A new algorithm for protein folding in the HP model*, In Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms, 876–884.

Alberts, B. Bray, D. Johnson, A. Lewis, J. Raff, M. Roberts, and Walter P. (1998), *Essential Cell Biology: An Introduction to the Molecular Biology of the Cell*, Garland Publishing Inc.

Carr, R., Hart, W. E., and Newman, A. (2004), *Bounding a protein free energy in lattice models via*

linear programming, Conference on Research in Computational Molecular Biology.

C. Rego, H. Li and F. Glover. (2006), *A filter-and-fan approach to the 2D HP model of the protein folding problem*, School of business administration, University of Mississippi.

Greenberg, H. J., Hart, W. E., and Lancia, G. (2004), *Opportunities for combinatorial optimization in computational biology*, INFORMS Journal on Computing, 16(3), 211–231.

H Yoon. (2006), *Optimization Approaches to Protein Folding*, PhD thesis, Georgia Institute of Technology.

Ken A. Dill. (1985), *Theory for the folding and stability of globular proteins*, Biochemistry, 24, 1501.

Ken A. Dill, Sarina Bromberg, Kaizhi Yue, Klaus M. Fiebig, David P. Yee, Paul D. Thomas, and Hue Sun Chan. (1995), *Principles of protein folding: A perspective from simple exact models*, Protein Science, 4, 561-602.

K. Yue and K. A. Dill. (1993), *Sequence-structure relationships in proteins and copolymers*, Physical Review E, 48(3), 2267–2278.

N. Lesh, M. Mitzenmacher, S. Whitesides. (2003), *A complete and effective move set for simplified protein folding*, Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology, 188-195.

P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis. (1998), *On the complexity of protein folding*, Journal of Computational Biology, 5(3), 423-466.

R. Backofen, S. Will. (2006), *A constraint-based approach to fast and exact structure prediction in three-dimensional protein models*, Journal of Constraints 11(1), 5–30.

Thang N. Bui and Gnanasekaran Sundarraj. (2005), *An efficient genetic algorithm for predicting protein tertiary structures in the 2D HP model*, Proceedings of the 2005 conference on Genetic and evolutionary computation, 305-392.

Vijay Candru, M. Rammohan Rao, and Ganesh Swaminathan. (2004), *The Sharpest Cut: The Impact of Manfred Padberg and His Work*, Society for Industrial Mathematics.

Vincenzo Cutello, Mario Pavone, Jonathan Timmis. (2007), *An Immune Algorithm for Protein Structure Prediction on Lattice Models*, IEEE Transactions On Evolutionary Computation, 11(1), 101-117.

William E. Hart and Sorin Istrail. (1996), *Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal*, Journal of Computational Biology, 3(1), 53-96.

William E. Hart and Alantha Newman (2006), *Handbook of computational molecular biology*, Chapman & Hall/CRC.

Xiaomei Li. (2007), *Protein Folding based on simulated annealing algorithm*, third international conference on natural computation, 4, 256-259.

Yu-Zhen Guo, En-Min Feng, and Yong Wang. (2007), *Optimal HP configurations of proteins by combining local search with elastic net algorithm*, Journal of biochemical and biophysical methods, 70, 335-340