**WORKING PAPER No.199**

# Protein Folding on Lattices: An Integer Programming Approach*

by

Vijay Chandru
M.Rammohan Rao &
Ganesh Swaminathan

September 2002

Please address all correspondence to:

Prof. M. Rammohan Rao
Professor
Indian Institute of Management Bangalore
Bannerghatta Road
Bangalore – 560076, India
E-mail: mrao@iimb.ernet.in
Phone : 080 – 6993136
Fax : 080 - 6584050

* Paper prepared for the Workshop held at Berlin during October 2001, to honour Prof. Padberg on his 60th Birthday. To appear in the edited book "The Sharpest Cut".

# Protein Folding on Lattices: An Integer Programming Approach

Vijay Chandru[*] M.Rammohan Rao[†] Ganesh Swaminathan[‡]

October, 2001 Revised January, 2002

**Abstract**

In this paper, we initiate the study of the protein folding problem from an integer linear programming perspective. The particular variant of protein folding that we examine is known as the hydrophobic-hydrophilic (HP) model of protein folding on the integer lattice. This problem is known to be NP-hard and also maxSNP-hard. We examine various alternate formulations for the planar version of this problem and present some preliminary computational results. Hopefully, this sets the stage for a polyhedral combinatorics assault on this important problem.

## 1 Introduction

Proteins are biological molecules that are responsible for implementing various functions in all living organisms. Each protein has well defined functions, which range from building up DNA and RNA molecules to controlling different parameters in living cells. It is amazing that proteins are built of very simple building blocks, known as amino acids [4]. There are twenty different amino acids. Amino acids are linked to each other by means of peptide bonds.

Determining the structure of proteins is a very important problem. The three dimensional structure of a protein is believed to be a very important determinant of the properties of the protein. This becomes crucial in drug design where the aim is to obtain proteins with specific functionalities. A remarkable discovery was made by Christian Anfinsen and his

[*] Indian Institute of Science, Strand Genomics, Bangalore. chandru@csa.iisc.ernet.in

[†] Indian Institute of Management, Bangalore. mrao@iimb.ernet.in

[‡] Strand Genomics, Bangalore. gans@strandgenomics.com

colleagues in the 1950s when they found that many simple proteins had a unique native structure, which just seems to depend on the sequence. This has been subsequently verified for a large number of proteins and it is now believed that the native structure is a minimum energy configuration (The Thermodynamic Hypothesis). This has led to an enormous interest in trying to develop methods to predict the three dimensional structure from the sequence information via optimization techniques. Determining a protein sequence has become feasible with current technology, but determining the exact three dimensional structure is still a very slow and expensive process that requires crystallization of the protein and a majority of proteins cannot be crystallized.

In principle, it should be possible to predict the fold of a protein into its native conformation, once we are given the sequence of the constituent amino acids. This is known as the protein structure prediction problem and is sometimes referred to as deciphering the second half of the genetic code. While large proteins fold in nature in seconds, computational chemists and biologists have found it to be a huge challenge to compute the minimum energy conformations using various formulations of this optimization problem. Recent work by theoretical computer scientists on this problem [8] has shown that the problems are NP-hard (cf. [13]) and even the very simple lattice model examined in this paper is known to be max-SNP hard and therefore unlikely to admit polynomial-time approximation schemes as well.

The difficulty of working with the detailed atomic level model has motivated biologists to work on simple discrete models. One way to discretize this problem is to only consider embeddings on a lattice. The energy function also has to be defined appropriately in this new setting. The resulting minimum energy conformation problems are essentially combinatorial optimization problems.

Broadly, three optimization modeling strategies have been proposed for protein folding.

**The Protein Structure Prediction Model (PSP model).** This model is a general nondiscrete model defined formally by Ngo and Marks [19], who also give a NP-hardness result for this model. In this model, the protein is described by the complete list of the atoms in the molecules, their connectivities, bond lengths and angles and force constants between all pairs of atoms. The energy of a conformation is a nonconvex function obtained by summing the contributions of different kinds of interactions. The NP-hardness is shown by a reduction from the partition problem.

**The Lattice Polymer Embedding Model (LPE model).** The LPE model was studied by Unger and Moult [27]. The protein is modelled as a chain of beads. The space is the collection of embeddings in the 3D cubic lattice. An embedding means that each bead must be placed at some lattice site, and successive beads must be adjacent on the lattice. In addition, the embedding must be not self-intersecting. The energy is defined as a weighted sum of pairwise interaction energies (functions that depend on the lattice distance between pairs of beads). The objective is to find the conformation that minimizes this energy. Unger and Moult show that this problem is NP-hard, by a reduction from the optimal linear arrangement. The HP model that we discuss later is a special case of this model.

**The Charged Graph Embedding Model (CGE model).** This model also describes the protein as a sequence of beads. A charge of -1, 0 or +1 is associated with each bead. For each pair of beads, the interaction energy is defined to be the product of the charges divided by the distance separating the beads (provided the distance is within a cutoff). The total energy is simply the sum of pairwise energies. One important condition is that bonds are allowed to cross, as long as there is at most one bead per site. Fraenkel [12] showed that this problem is also NP-hard by reduction from 3D matching. The CGE model incorporates charges on the residues, which is a realistic feature; but the bonds permitted are too general.

We now consider a popular model of protein folding called the Hydrophobic-Hydrophilic model.

**The Hydrophobic-Hydrophilic Model (HP model).** This model was introduced by Dill [10] as a special case of the LPE model and has been studied extensively in [11,16,17,18, 28,29] and is the simplest possible abstraction of the folding problem, which is still nontrivial and retains the hardness features of the original problem.

The model starts with classifying the twenty amino acids as H (Hydrophobic or nonpolar) and P (Hydrophilic or polar). This classification is known from experimental results. A protein is modelled as a sequence of H's and P's. The conformations allowed are not self-intersecting embeddings on a two or three dimensional cubic lattice. A pair of amino acids that occur in successive positions in the chain are called *connected neighbours, while a pair of nonsuccessive amino acids that are adjacent in the embedding are called* topological neighbours. The energy of any folding is proportional to the negative of the number of pairs of H's that are topological neighbours. Therefore, the aim is to maximise the number of topological neighbours.

Even this simple model is NP-hard to solve, and proving this was an open question for a long time (see [1,2,3,8,9,15,20,22,26]). Even before hardness results were known, Hart and Istrail [14] gave a simple approximation algorithm, which achieves a worst-case ratio of 1/4 for 2D lattices and 3/8 for the 3D case. Very recently, Alantha Newman [21] has improved the 1/4 bound for the 2D case to a 1/3 bound with a linear-time approximation algorithm. A lot of empirical work has also been done on this model. Dill et al. [11] have extensively studied the biological properties of this model by actual enumeration of all conformations for small length sequences. Unger and Moult [28,29] looked at this problem from a genetic algorithm viewpoint and they were able to obtain compact foldings of fairly long sequences, but they were not able to give any guaranteed bounds on their algorithms.

In this paper we focus on the HP model and in particular on the 2D lattice embedding of the main chain. The next five sections describe integer programming formulations of this problem. We report some very preliminary computational experiments carried out on these formulations in Section 7 and conclude with a brief agenda for research on folding proteins using integer programming.

## 2 Formulation

The 2D - HP protein folding model on a rectangular lattice is formulated as an integer linear programming problem. A protein is a chain of amino acid residues. The sequence of amino acids in the chain to be folded on the two-dimensional grid is denoted as $s_k$, $k = 1, 2, \cdots, n$.

Each amino acid $s_k$ is either hydrophobic or hydrophilic. The set of amino acids that are hydrophobic is denoted as $H$. Amino acids $s_t$ and $s_{t+1}$, $1 \leq t \leq n-1$ are adjacent on the chain.

In this formulation, a $(2n-1) \times (2n-1)$ grid is used. Each lattice point or vertex is denoted as $(i, j)$, $1 \leq i, j \leq 2n-1$. Two vertices $(i, j)$ and $(u, v)$ are said to be neighbours on the grid if one of the following holds.

- $u = i$ and $v = j+1$ or $v = j-1$
- $v = j$ and $u = i+1$ or $u = i-1$

The set of vertices adjacent to vertex $(i,j)$ is denoted as $N_{ij}$. Note that if $(u,v) \in N_{ij}$, then $(i,j) \in N_{uv}$. We define the *grid graph* $G = (V,E)$, where every edge $e$ is of the form $((i,j),(u,v))$ where $(u,v) \in N_{ij}$ and $1 \leq i,j \leq 2n-1$. The first amino acid, $s_1$, is assumed to be anchored at the centre of the grid, i.e, at the lattice point $(n,n)$. In Section 4, it is shown that the size of the grid can be reduced considerably, thereby eliminating a large number of variables.

The protein folding problem on a two-dimensional grid involves placing the amino acids $s_k$, $1 \leq k \leq n$ at the vertices $(i,j)$, $1 \leq i,j \leq 2n-1$ such that the following constraints are satisfied.

(i) Each amino acid is placed at precisely one vertex.

(ii) Each vertex has at most one amino acid.

(iii) Amino acids that are adjacent on the chain must be placed at adjacent vertices.

The objective is to place the amino acids on the vertices so that a maximum number of amino acids in the set H that are nonadjacent on the chain are adjacent on the grid, i.e., are topologically adjacent.

The variables are defined as follows: for $1 \leq i,j \leq 2n-1$ and $1 \leq k \leq n$,

- $x_{ij}^k$ is 1 if amino acid $s_k$ is placed at the grid point $(i,j)$ and 0 otherwise.
- $y_{ij}^{uv}$ is 1 if some $s_a \in H$ and $s_b \in H$ are placed at the vertices $(i,j)$ and $(u,v)$ which are neighbours, i.e., $(u,v) \in N_{ij}$, and 0 otherwise

The integer programming formulation is as follows:

$$(P) \qquad \max \sum_{((i,j),(u,v)) \in E} y_{ij}^{uv}$$

subject to

$$x_{nn}^1 = 1 \tag{1}$$

$$\textstyle\sum_{k=1}^n x_{ij}^k \leq 1 \text{ for } 1 \leq \text{i,j} \leq 2\text{n} - 1 \tag{2}$$

$$\textstyle\sum_{i=1}^{2n-1} \sum_{j=1}^{2n-1} x_{ij}^k = 1 \text{ for } 1 \leq k \leq n \tag{3}$$

$$x^k_{ij} \leq \sum_{(uv)\in N_{ij}} x^{k+1}_{uv} \text{ for } 1 \leq i,j \leq 2n-1,\ 1 \leq k \leq n-1 \tag{4}$$

$$y^{uv}_{ij} \leq \sum_{k\in H} x^k_{ij} \text{ for all edges } ((i,j),(u,v)) \tag{5}$$

$$y^{uv}_{ij} \leq \sum_{k\in H} x^k_{uv} \text{ for all edges } ((i,j),(u,v)) \tag{6}$$

$$x^k_{ij} = 0 \text{ or } 1 \text{ for } 1 \leq i,j \leq 2n-1 \text{ and } 1 \leq k \leq n \tag{7}$$

$$y^{uv}_{ij} \geq 0 \text{ for all edges } ((i,j),(u,v)) \tag{8}$$

The first constraint anchors the first amino acid, $s_1$, in the chain. The second set of constraints ensures that at most one amino acid is placed at any vertex. The requirement that each amino acid is placed at some vertex is ensured by the constraint set (3).

The first amino acid is anchored at the vertex $(n,n)$ and the constraint in (4) for $i=n$, $j=n$ and $k=1$ ensures that the second amino acid is placed at a vertex, say $(a,b) \in N_{nn}$ . Next the constraint in (4) for $i=a$, $j=b$ and $k=2$ ensures that the third amino acid is placed at a vertex in $N_{ab}$. Repeating this argument, it follows that all amino acids are placed at some vertex. However constraint set (3) ensures that no amino acid is placed at more than one vertex.

The constraint sets (5) and (6) imply that $y^{uv}_{ij}$ may be set to 1 only if an amino acid $s_a \in H$ is placed at vertex $(i,j)$ and another amino acid $s_b \in H$ is placed at the neighbouring vertex $(u,v)$. Because of the objective function, it follows that $y^{uv}_{ij}$ is set to 1 if and only if two hydrophobic amino acids are placed at neighbouring vertices $(i,j)$ and $(u,v)$. The constraint sets (5) and (6) are written in a convenient form. Clearly, some constraints are duplicated since $(u,v) \in N_{ij}$ implies $(i,j) \in N_{uv}$. It is understood that such duplicates are eliminated, i.e., for any pair $(i,j)$ and $(u,v)$ of neighboring vertices only one constraint in (5) and one constraint in (6) are required.

Constraint set (7) ensures integrality of the variables $x^k_{ij}$. The variables $y^{uv}_{ij}$ are restricted to be nonnegative variables in (8). It is not necessary to require that the variables $y^{uv}_{ij}$ are 0 or 1. This is because of the integrality restriction on $x^k_{ij}$ and the objective function involves maximizing the sum of the variables $y^{uv}_{ij}$.

The objective function maximizes the number of hydrophobic amino acids that are placed at adjacent vertices. The optimal objective function value includes a constant which is the number of hydrophobic amino acids that are adjacent in the chain. This is because the true objective is to maximize the number of hydrobhic topological neighbors while the objective in the formulation counts the number of hydrophobhic neighbors both topological and adjacent on the chain.

**Remark 2.1** It is straightforward to extend the formulation to other lattices such as triangular lattices or 3 dimensional lattices. The size of the lattice and the set $N_{ij}$ change accordingly and additional restrictions, if required, can be easily imposed.

It is also possible to include interactions between amino acids that are placed on nonadjacent vertices on the lattice but within some specified distance. This requires additional variables $y_{ij}^{uv}$ where vertex $(u, v)$ is within the specified distance from vertex $(i, j)$. The formulation can easily be extended to the case of generalized hydrophobicity that is discussed in [2]. In this case, the amino acids $s_k$, $1 \leq k \leq n$ in the chain are not restricted to be only hydrophobic or hydrophilic but can be any one of the 20 possible amino acids. The energy between two topological neighbours can be an arbitrary function that depends upon the type of amino acids. This extension of the problem requires that the variables $y_{ij}^{uv}$ must now have two additional parameters, say $a$ and $b$ denoting the amino acids that are not adjacent on the chain. Then the constraints (5) and (6) are to be replaced by

$$y_{ij}^{uv}(a,b) \leq x_{ij}^{a}, \;\; 1 \leq i,j \leq 2n-1, \;\; 1 \leq a,b \leq n \tag{9}$$

$$y_{ij}^{uv}(a,b) \leq x_{uv}^{b}, \;\; 1 \leq i,j \leq 2n-1, \;\; 1 \leq a,b \leq n \tag{10}$$

where $a$ and $b$ are nonadjacent in the chain and $(u, v) \in N_{ij}$.

The objective function coefficient for $y_{ij}^{uv}(a,b)$ would be the energy between amino acids $s_a$ and $s_b$ that are not adjacent on the chain but adjacent on the grid. It is understood that if the energy between topologically adjacent amino acids $s_a$ and $s_b$ is zero, the corresponding variables and constraints in (9) and (10) are eliminated from the formulation.

## 3 Additional Inequalities

Instead of merely restricting the position of the neighbouring amino acids $s_k$ and $s_{k+1}$, we can write constraints that restrict the position of amino acids $s_k$ and $s_{k+t}$ where $t \geq 1$.

A path (or a simple path) between two vertices $((i, j), (u, v))$ in the graph G is defined in the usual manner as a sequence of edges connecting vertices $(i, j)$ and $(u, v)$ with no intermediate vertex repeated. It follows that the shortest distance between any pair of nodes $(i, j)$ and $(u, v)$ is $d_{ij}^{uv} = |u - i| + |v - j|$.

Then we have the constraints

$$x_{ij}^{k} \leq \sum_{(u,v):d_{ij}^{uv} \leq t} x_{uv}^{k+t} \text{ for } 1 \leq i, j \leq 2n-1;\; 1 \leq k \leq n-t \tag{11}$$

Viewing the chain of amino acids in the reverse direction, analogous to constraint set (11), we have for $t \geq 1$ the constraints.

$$x_{ij}^{k} \leq \sum_{(u,v):d_{ij}^{uv} \leq t} x_{uv}^{k-t} \text{ for } 1 \leq i, j \leq 2n-1;\; t+1 \leq k \leq n \tag{12}$$

These constraints ensure that if amino acid $s_k$ is placed at vertex $(i,j)$, then amino acid $s_{k-t}$ must be placed at a vertex which is at a distance less than or equal to $t$ from vertex $(i,j)$.

Integrality of the variables $x_{ij}^{k}$ and the constraint set (4) imply the constraints in (11) and (12). However some fractional solutions to (P) are cut off by (11) and (12). The usefulness of these constraints from a computational point of view needs to be explored. The inequalities

$$\sum_{k \in H} x_{ij}^{k} + \sum_{k \in H} x_{uv}^{k} \leq 1 + y_{ij}^{uv} \text{ for all edges } ((\mathrm{i.j}),(\mathrm{u,v})) \in \mathrm{E} \tag{13}$$

are easily verified to be valid. The formulation (P) needs to be studied from a polyhedral combinatorics perspective. For example, it would be interesting to identify the facet-defining inequalities, if any, in the above set of valid inequalities

## 4 Grid Size and Elimination of Variables

In the formulation (P) in Section 2, it is possible to set a number of variables to zero, i.e., to eliminate several variables. This results in a more compact formulation and more importantly, can help partial enumeration based optimization algorithms by pruning off the search space.

The shortest distance between any pair of vertices is either even or odd. It is easy to verify that if the shortest distance between any pair of vertices $(i,j)$ and $(u,v)$ is even (odd), then all paths between the same two vertices are of even (odd) length.

The distance between two amino acids $s_a$ and $s_b$ in the chain is defined as $w_{ab} = |a-b|$. Clearly, the distance between two amino acids is either even or odd. It follows that two amino acids which are at even (odd) distance in the chain must be at even (odd) distance on the grid. Moreover, two amino acids $s_a$ and $s_b$ with distance $w_{ab}$ can never be placed at vertices $(i,j)$ and $(u,v)$ where $d_{ij}^{uv} > w_{ab}$. Note, however, that it is possible to place $s_a$ and $s_b$ at vertices $(i,j)$ and $(u,v)$ where $d_{ij}^{uv} < w_{ab}$.

Now, noting that the first amino acid, $s_1$, is anchored at vertex $(n,n)$ it follows that the second amino acid $s_2$ can be placed only at a vertex which is at a shortest distance of 1 from vertex $(n,n)$, i.e, at vertex $(n-1,n)$, $(n+1,n)$, $(n,n-1)$, or $(n,n+1)$. Similarly, the third amino acid $s_3$, can be placed only at a vertex which is at a shortest distance of two. Moreover, since $w_{13}$ is even, $s_3$ cannot be placed at a vertex $(u,v)$ whose shortest distance from vertex $(n,n)$ is odd. Continuing, the fourth amino acid $s_4$, can be placed only at a vertex which is at a shortest distance of one or three but not more, i.e since $w_{14}$ is odd, $s_4$ can be placed at a vertex $(u,v)$ such that $d_{nn}^{uv}$ is odd and less than or equal to 3.

By repeating the above argument for amino acids $s_i$, $5 \leq i \leq n$ a large number of variables can be eliminated from the formulation. It should be noted that some vertices $(u,v)$ in the grid such that $d_{nn}^{uv} > n$ can be eliminated from the grid itself. Eliminating all such variables, it follows that when $n$ is odd, the number of $x_{ij}^k$ variables is $\frac{2}{3}p(p+1)(4p+5)$ where $p = \frac{n-1}{2}$. In this case, the number of $y_{ij}^{uv}$ variables is $4(n-1)^2$.

It is also possible to reduce the size of the grid itself by noting that any folding of the protein can be rotated as necessary. Suppose, as before, we anchor the first amino acid at vertex $(n,n)$. Let $p = \lfloor \frac{n}{2} \rfloor$ where $\lfloor y \rfloor$ denotes the largest integer less than or equal to $y$. It suffices to consider the rectangular lattice with vertices $(i,j)$, $n-p \leq i \leq 2n-1$; $n-p \leq j \leq n+p$. In this rectangular grid also, there are some vertices $(u,v)$ such that $d_{nn}^{uv} > n$. Such vertices can be eliminated from the grid.

## 5 Alternate Formulation

Instead of anchoring the first amino acids $s_1$ at vertex $(n,n)$, the first amino acid may be placed anywhere. In this case, we require only a $n \times n$ grid to begin with. Now we need the constraint

$$\sum_i \sum_j x_{ij}^1 = 1 \tag{14}$$

to ensure that the first amino acid is placed at some vertex. Moreover, in this formulation, we cannot eliminate the variables as in Section 4. However by increasing the grid size to $(n+1) \times (n+1)$ it is possible to restrict the placing of the first amino acid $s_1$ to be at a vertex $(i,j)$ where $i$ and $j$ are odd. Given this restriction, it is easily verified that amino acid $s_2$ can be placed only at a vertex $(u,v)$ such that one of the following holds

$$u \text{ is odd and } v \text{ is even} \tag{15}$$

$$u \text{ is even and } v \text{ is odd} \tag{16}$$

Continuing, it follows that amino acid $s_3$ can be placed only at a vertex $(a, b)$ such that one of the following holds:

$$a \text{ is odd and } b \text{ is odd} \tag{17}$$

$$a \text{ is even and } b \text{ is even} \tag{18}$$

It is easily verified that amino acids which are at an odd distance from $s_1$ in the chain may be placed at a vertex $(u, v)$ that satisfies either (15) or (16) while amino acids which are at an even distance from $s_1$ in the chain may be placed at a vertex $(a, b)$ that satisfies either (17) or (18).

By this argument, a large number of variables can be eliminated in this alternate formulation. Eliminating all such variables, it follows that when $n$ is odd, the number of $x_{ij}^k$ variables is $\frac{1}{4}(n+1)^2(2n-1)$ while the number of $y_{ij}^{uv}$ variables is $2n(n+1)$.

The alternate formulation is analogous to the formulation in Section 2 except that the grid size is different and constraint (1) is now replaced by constraint (14).

## 6 Row and Column Generation

A feasible solution to (P) has exactly $n$ of the variables $x_{ij}^k$ equal to 1. Typically in a feasible solution only a few (perhaps of order $n$) of the variables $y_{ij}^{uv}$ are equal to 1. However, inspite of eliminating several variables as indicated in Sections 4 and 5, the formulation in Section 2 and the alternate formulation in Section 5 have a large number of variables and constraints. The number of variables and constraints are of the order $n^3$. Hence, feasible solutions to (P) are highly degenerate.

In order to speed up the computations, it might be desirable to start with a small number of constraints and variables. Given an optimal solution to the linear programming relaxation of the smaller problem, it is straight forward to generate, if there is one, a violated constraint from among those not written down explicitly. Similarly, it is straight forward to generate from among those variables which have not been written down explicitly, a variable if there is one, to enter the basis. Thus, to keep the size of the working basis small and thereby speed up the computations, it is possible to resort to row and column generation.

The process of row and column generation is akin (though not identical) to starting with a thin rectangular lattice and solving the problem repeatedly by increasing the width and length of the lattice.

## 7 Computational Results

To determine the folding of a protein consisting of $n$ amino acids, the current (alternate) formulation uses a grid of $n^2$ lattice points. Since any of the $n$ amino acids can occupy any of the $n^2$ lattice points, the number of variables $x_{ij}^k$ are $n^3$. The number of $y_{ij}^{uv}$ variables is $2n(n-1)$. However, the maximum size of the lattice spanned by a protein of length $n$ consists of $p$ rows and $k$ columns such that $p+k \leq n+1$. This can be utilized to reduce the size of the problem. The following outlines the approach that has been used for computational purposes.

1. Initialize $p = 2$.

2. Use a grid of size $p \times k$ such that$p + k = n + 1$.

3. Calculate the optimal objective function value for the associated integer programming problem.

4. Increment $p$ by 1. If $p = n$, go to step (5) else go to step (2).

5. Calculate the maximum of the optimal objective function values obtained in the above $n - 2$ iterations.

If $n-2$ iterations (the value of $p$ varies from 2 to $n-1$) are done, then one of the solutions will be an optimal solution to the given protein folding problem. Further by considerations of symmetry, one can vary $p$ from 2 to $\lfloor \frac{n}{2} \rfloor$, leading to $\lfloor \frac{n}{2} \rfloor - 1$ iterations. For a $p \times k$ lattice the number of $x_{ij}^k$ variables is $pkn$ and the number of $y_{ij}^{uv}$ variables is $2pk - p - k$. The total time taken for $\lfloor \frac{n}{2} \rfloor - 1$ iterations with reduced lattice points can be expected to be less than the time taken for one iteration with $n^2$ lattice points. That is what we observed in our limited computational results.

The following is an example illustrating the results of the approach outlined above. The protein has 10 amino acids with three hydrophobic amino acids at positions 1,4 and 7. For

Table 1: Partial Enumeration

| Lattice<br>size | Opt Int<br>Obj Value | Total number of<br>Simplex iterations | Number of nodes<br>in B&B Tree |
|---|---|---|---|
| $2 \times 9$ | 1 | 13921 | 222 |
| $3 \times 8$ | 2 | 2805 | 39 |
| $4 \times 7$ | 2 | 5028 | 49 |
| $5 \times 6$ | 2 | 16401 | 131 |

Table 2: Optimality Gaps

| Problem<br>Number | $n$ | Position of<br>H amino acids | Opt Value<br>LP relax | Opt Value<br>IP | Obj Val<br>HI-heuristic |
|---|---|---|---|---|---|
| 1 | 10 | 1,4,7 | 5.4 | 2 | 1 |
| 2 | 10 | 1,4,7,8 | 6.2 | 3 | 2 |
| 3 | 11 | 1,4,7,8,11 | 8.09 | 4 | 2 |
| 4 | 11 | 1,3,7,8 | 6.27 | 2 | 1 |

different lattice sizes, the optimal objective function value for the integer programming problem, the number of simplex iterations that were required and the number of nodes generated in the branch and bound scheme are summarized in Table 1.

Four problems were solved using the integer programming (IP) approach and the Hart-Istrail (HI) heuristic. The number of amino acids ($n$) in the chain, the positions of the hydrophobic amino acids, the optimal objective function value obtained by IP and the objective function value obtained by HI heuristic are given in the table below. The results clearly show that the Hart-Istrail heuristic produces solutions that are far from being optimal in relative terms.

The IP approach uses the $n \times n$ grid. The optimal objective function values for the linear programming (LP) relaxation and the IP approach that are given in Table 2 are after subtracting the constant given by the number of hydrophobic amino acids that are adjacent in the chain.

In all the above problems, none of the variables, as suggested in Section 5, were eliminated. It is planned to implement those improvements in the near future.

## 8 Conclusion

The lattice models of protein folding have certain limitations [11].

- The resolution of the original problem is lost. Bond angles actually lie in some restricted regions as indicated by the Ramachandran plot, rather than being right angles.
- Details of protein structure, bond energies and charges cannot be represented accurately in such models.
- Bond lengths are not captured well enough.

An attempt to address the first limitation has been to explore alternate lattice structures such as triangular lattices[2]. In spite of the above limitations, discrete lattice models continue to be useful particularly for simulations and for enumerative techniques for deducing statistical properties of small proteins and peptides [5,6,7.23,24,25]. An important challenge is therefore to improve the integrity of lattice models.

The computational results reported in Section 7 are clearly at a very preliminary stage. We intend to continue with the experimentation and will report our findings in a subsequent paper.

**Acknowledgement.** The authors would like to thank Professor Ramesh Hariharan for some early discussions of the integer programming approach and in particular for the idea of incrementally expanding lattices which led us to the interpretation in Section 6. They would also like to thank an anonymous referee for suggesting the valid inequalities (13).

## References

[1] J. Atkins and W.E. Hart. On the Intractability of Protein Folding with a Finite Alphabet of Amino Acids, *Algorithmica*, 25. pp 279-294, 1999.

[2] R. Agarwala, S. Batzogloa, V. Dancik, S.E. Decatur, S. Hannenhalli, M. Farach, S. Muthukrishnan and S. Skiena. Local rules for protein folding on a triangular lattice and generalised hydrophobicity, *RECOMB*, pp 1-2, 1997.

[3] B. Berger and T. Leighton. Protein Folding in the hydrophobic-hydrophilic model is NP complete, *J. Comput. Biol.*, 5, pp 27-40, 1998.

[4] C. Brandon and J. Tooze. *Introduction to Protein Structure*, Second Edition, Garland, 1999.

[5] H.S. Chan and K.A. Dill. Origins of structure in globular proteins, *Proc. Natl. Acad. Sci. USA*, 87, pp 6388-6392, 1990.

[6] H.S. Chan and K.A. Dill. Polymer principles in protein structure and stability, *Ann. Rev. of Biophysics and Biophysical Chem.*, 20, pp 447-490, 1991.

[7] P. Clote. Protein folding, the Levinthal paradox and rapidly mixing Markov chains, *ICALP*, pp 240-249, 1999.

[8] V. Chandru, A. Dattasharma, V.S. Kumar, The Algorithmics of Folding Proteins on Lattices, to appear in *Discrete Applied Mathematics*, 2002.

[9] P. Crescenzi, D. Goldman, C. Papadimitrou, A. Piccolboni, M. Yannakakis. On the complexity of Protein Folding, *J. Comput. Biol.*, 5, 423-446, 1998.

[10] K.A. Dill. *Biochemistry*, 24, pp 1501, 1985.

[11] K.A. Dill, S. Bromberg, K. Yue, K.M. Fiebig, D.P. Yee, P.D. Thomas and H.S. Chan. Principles of Protein Folding: a perspective from simple exact models, *Protein Sci.*, 4, 561-602, 1995.

[12] A.S.Fraenkel. Complexity of protein folding. *Bull. Math. Bio.*, 1993.

[13] M.R. Garey and D.S. Johnson. *Computers and Intractability- A Guide to the theory of NP-completeness*, Freeman, San Francisco, CA 1979.

[14] W.E. Hart and S. Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eights of optimal, *J. Comput. Biol.*, 3, pp 53-96, 1996.

[15] V. Heun. Folding, *Proc. of the Euro. Symp. Alg.*, 1999.

[16] K.F. Lau and K.A. Dill. A Lattice statistical mechanics model of the conformation and sequence spaces of proteins, *Macromolecules*, 22, pp 3986-3997, 1989.

[17] K.F. Lau and K.A.Dill. Theory for protein mutability and biogenesis, *Proc. Natl. Acad. Sci. USA*, 87, pp 638-642, 1990.

[18] D. Lipman and J. Wilber. *Proc. Royal Soc. London*, 245, 1991.

[19] J.T.Ngo and J.Marks. Computational complexity of a problem in molecular-structure prediction, *Protein Engineering*, 5(4), pp 313-321, 1992.

[20] A. Nayak, A. Sinclair and U. Zwick. Spatial codes and the hardness of string folding problems, *Proc. of 9th ACM-SIAM Symp. on Disc. Algo.*, pp 639-648, 1998.

[21] A. Newman, A new algorithm for protein folding in the HP Model, *Symposium on Discrete Algorithms* ACM/SIAM, 2002.

[22] M. Paterson and T.Przytycka. On the complexity of string folding, *Disc. App. Math.*, 71, pp 217-230, 1996.

[23] A. Sinclair. *Algorithms for random generation and counting: A Markov chain approach*, Birkhauser, 1993.

[24] A. Sali, E. Shakhnovich and M. Karplus. How does a protein fold?, *Nature*, 369, pp 248-251, 1994.

[25] A. Sali, E. Shakhnovich and M. Karplus. Kinetics of protein folding: A lattice model study of the requirements for folding to the native state, *J. Mol. Bio.*, 235, pp 1614-1636, 1994.

[26] L. Trevisan. When Hamming meets Euclid: The approximability of geometric TSP and MST, *Proc. 29th ACM Symp. on the Theory of Comput.*, pp 21-29, 1997.

[27] R.Unger and J.Moult. Finding the lowest free energy conformation of a protein is a NP-hard problem: Proof and implications, *Bull. Math. Bio.*, 1993.

[28] R. Unger and J. Moult. Genetic algorithms for protein folding simulations, *J. Mol. Bio.*, 231, pp 75-81, 1993.

[29] R. Unger and J. Moult. A genetic algorithm for three dimensional protein folding simulations, *Proc. 5th Int. Conf. Genetic Algo.*, pp 581-588, 1993.