Application of Constraint Programming Techniques for Structure Prediction of Lattice Proteins with Extended Alphabets

Rolf Backofen, Sebastian Will
Institut für Informatik, LMU München
Oettingenstraße 67, D-80538 München
and
Erich Bornberg-Bauer
Bioinformatics Group, European Media Laboratory,
Schloß-Wolfsbrunnenweg 33, D-69118 Heidelberg.

February 24, 1999

Abstract

Predicting the ground state of biopolymers is a notoriously hard problem in biocomputing. Model systems, such as lattice proteins are simple tools and valuable to test and improve new methods. Best known are HP-type models with sequences composed from a binary (hydrophobic and polar) alphabet. Major drawback is the degeneracy, i.e. the number of different ground state conformations.

Here we show how recently developed constraint programming techniques can be used to solve the structure prediction problem efficiently for a higher order alphabet. To our knowledge it is the first report of an exact and computationally feasible solution to model proteins of length up to 36 and without resorting to maximally compact states. We further show that degeneracy is reduced by more than one order of magnitude and that ground state conformations are not necessarily compact. Therefore, more realistic protein simulations become feasible with our model.

Abbreviations: HP - hydrophobic, polar; CP - Constraint-Programming, HPNX - hydrophobic, positive, negative, neutral; MCS - maximum compact state

1 Introduction

The protein structure prediction is one of the most important unsolved problems of computational biology. It can be specified as follows: Given a protein by its sequence of amino acids, what is its native structure? NP-hardness has been proved for many different models (including lattice and off-lattice models). These results strongly suggest that the protein folding problem is NP-hard in general. Therefore, it is unlikely that a general, efficient algorithm for solving this problem can be given. Actually, the situation is even worse, since the general principles why natural proteins fold into a native structure are unknown. This is cumbersome since rational design is commonly viewed to be of paramount importance e.g for drug design. One problem is that artificially designed proteins usually don't have a unique and stable native structure.

To tackle this problem simplified models have been introduced. They have became a major tool for investigating general properties of protein folding. Most important are the so-called lattice models. The simplifications commonly used in this class of models are: 1) monomers (or residues) are represented using a unified size 2) bond length is unified 3) the positions of the monomers are restricted to lattice positions and 4) a simplified energy function.

In principle, one can approximate real proteins arbitrarily close using sufficiently complex lattice models. While highly connected lattices were used primarily for simulating folding of real-sized proteins [25, 21], square and cubic lattices were preferred to study basic principles. The HP (hydrophobic-polar) model [18, 12] is definitely the model of utmost simplicity since it models exclude volume, hydrophobicity and conformational flexibility while all other protein-like properties are ignored. Essentially it is a polymer chain representation on a lattice with exactly one stabilizing interaction when two hydrophobic residues are neighbors on the lattice but not along the chain. This enforces compactification while polar residues and solvent is not explicitly regarded. It follows the assumption that the hydrophobic effect determines the overall configuration of a protein. Its major drawback is certainly the crude energy potential which results in a very rugged energy landscape and, especially in 3-dimensional models, a considerable amount of degeneracy [30]. This means that the lowest energy state is not a single structure but has many different conformations.

Other models use energy parameters derived form the random energy model or experimentally determined potentials such as the Miyazawa-Jernigan contact potential. These models certainly have the appeal that results are energetically comparable to real proteins and a more realistic folding behavior. However, to enable tractability, computations must be restricted to maximum compact shapes and very small fractions of sequence space (i.e. unique sequences). Examples of how such models can be used for predicting the native structure or for investigating principles of protein folding were given [28, 1, 13, 27, 15, 2, 20] Mostly very attractive

potentials (with a shifted mean) are used and all possible MCS (maximum compact states) configurations (ca. 10^5 on a $3 \times 3 \times 3$ cube), are exhaustively tested for each sequence. Folding experiments are in general performed using Monte Carlo techniques. Typically one finds the native conformation within 50 000 000 Monte Carlo steps. In performing such experiments, it is clear that the quality of the predicted principle depends on several parameters. The first is the quality of the used lattice and energy function. The second, and even more crucial point, is the ability for finding the native structure. For the energy function used by [28], there is no *exact* algorithm for finding the minimal structure. To be computationally feasible, the search for the native structure was restricted to a $3 \times 3 \times 3$-cube. But this approach has some drawbacks, some of them were previously pointed out in [8]: 1) The energy function had to be biased to a mean hydrophobicity in order to get proteins whose native structure is on the $3 \times 3 \times 3$-cube with high probability (see [28]); 2) even then, it is not guaranteed that the minimal conformation is on this cube. Examples for the HP-model have been shown in [30]; 3.) the length of the proteins cannot be arbitrarily chosen. Since there is an algorithm for finding the native structure on the HP-model, one could think of redoing the experiment within the HP-model. But the HP-model has the problem that its degeneracy (i.e., the number of structures of a sequence that have minimal energy) is large [12, 30]. Hence, there is no dedicated native structure. For this reason, extended models such as the HPNX-model (HPNX=hydrophobic, positive, negative, neutral) [6] have been introduced.

Other groups have studied the principles of the sequence space to shape space mapping in an evolutionary context and the influence of mutations on possible fitness values associated to these structures [14, 15, 7]. A detailed discussion of lattice proteins in general can be found in [12].

Recent progress in CP (constraint programming) [26] has made it possible to apply straightforward techniques to predict the global minimum of proteins [4]

The main point of this paper is to demonstrate the applicability of CP to more complex, realistic problems. While formal details of the specific algorithm must be omitted because of space constraints and are described in another paper [4] the goal of this contribution is twofold: firstly, we demonstrate that it is possible to convey a solvable model for lattice proteins which is neither restricted to MCS nor small alphabets. Secondly, we demonstrate the advantages of using such an alphabet since degeneracies are drastically reduced. Finally, we explicitly show and discuss some examples for these achievements in lattice protein folding.

2 Constraint Programming for higher Alphabets

2.1 Lattice proteins

Our studies are based on the HP-model, which has been introduced by [18, 19]. In this model, the 20 letter alphabet of amino acids (and the corresponding manifoldness of forces between them) is reduced to a two letter alphabet, namely H and P. H represents *hydrophobic* amino acids, whereas P represent *polar* or hydrophilic amino acids. The energy function for the HP-model is given by the matrix as shown in Figure 1(a). It simply states that the energy contribution of a contact between two monomers is -1 if both are H-monomers, and 0 otherwise. Two monomers form a *contact* in some specific conformation if they are not connected via a bond, but occupy neighboring positions in the conformation (i.e., the euclidian distance of the positions is 1). A conformation with *minimal energy* (in the following called *optimal conformation*) is just a conformation with the maximal number of contacts between H-monomers. Just recently, the structure prediction problem has been shown to be NP-complete even for the HP-model [5, 10].

(a)

	H	P
H	-1	0
P	0	0

(b)

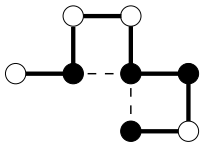


Figure 1: Energy matrix and sample conformation for the HP-model

A sample conformation for the sequence PHPPHHPH in the two-dimensional lattice with energy -2 is shown in Figure 1(b). The white beads represent P, the black ones H monomers. The two contacts are indicated via dashed lines. It can be seen that the very first residue has alternative positions without changing the energy, i.e. the ground state is "degenerate".

In the following we will describe how the principles of constraint programming can be applied to higher order alphabets. In principle, specific rules must be applied for every alphabet. Ideally one would use experimentally derived potentials, such as Crippen's 4-letter alphabet [11] which was, however, derived for a square lattice. Because of the before mentioned problems with larger alphabets with potentials derived from real proteins we use the ad-hoc HPNX potential from [6]. It can be seen as an intuitive extension of the HP potential. It is our goal to show that CP is well able to cope with different strengths of interaction and repulsions as well (repulsions were repeatedly argued to be of great importance for a unique solution to the structure prediction problem [9, 24]). The HPNX-model is an extension of the HP-model where the polar monomers are split into positively charged (P), negatively charged (N) and neutral (X) monomers. The energy function of the HPNX-model

is given by the matrix

	H	P	N	X
H	-4	0	0	0
P	0	1	-1	0
N	0	-1	1	0
X	0	0	0	0

(1)

2.2 Basic Constraints and Search Algorithm

We start with the basic constraint formulation that underlies our search algorithm. This formulization is straightforward, but we have added to show how the problem can be defined in Constraint-Programming. Clearly, this basic formulation is not sufficient to yield an efficient search algorithm. But it shows how the constraint-based search can be used predicting a minimal energy structure of an HP (resp. HPNX) sequence. We then indicate which constraints have to be added and how to modify the search strategy in order to yield an efficient algorithm.

Our algorithm is based on constraint optimization, which is the combination of two principles, namely generate-and-constraint with branch-and-bound. For using constraint optimization, we have to transform the structure prediction problem into a constraint problem. A constraint problem consists of a set of variables together with some constraints (relations) on these variables.

For specifying the basic constraint problem, we need some definitions. We will describe the constraint formulation for the HP-model. Since the basic constraint formulation is the same for the HP- and the HPNX-model, we will talk of polar monomers meaning P-monomers in the HP-model and PNX-monomers in the HPNX-model.

Let $s = s_1 \dots s_n$ be an HP- (or HPNX)-sequence of length n. A conformation c for this sequence is nothing else but a function $c : [1..n] \mapsto \mathbb{Z}^3$ assigning vectors to monomers such that

1. for all $1 \leq i < n$ we have $\|c(i) - c(i+1)\| = 1$ (i.e., every two successive monomers i and $i+1$ have distance 1)

2. and for all $i \neq j$ we have $c(i) \neq c(j)$ (the conformation c is self-avoiding).

Now we can encode the space of all possible conformations for a given sequence as a constraint problem as follows. We introduce for every monomer i new variables $\mathtt{X}_i$, $\mathtt{Y}_i$ and $\mathtt{Z}_i$, which denote the x-, y-, and z-coordinate of $c(i)$. Since we are using a cubic lattice, we know that this coordinates are all integers. But we can even restrict the possible values of these variables to the finite domain $[0..2n]$.[1] This is

[1] We even could have used $[1..n]$. But the domain $[0..2n]$ is more flexible since we can assign an arbitrary monomer the vector (n, n, n), and still have the possibility to represent all possible conformations.

expressed by introducing the constraints

$$\mathtt{X}_i \in [1..(2 \cdot \text{length}(s))] \wedge \mathtt{Y}_i \in [1..(2 \cdot \text{length}(s))] \wedge \mathtt{Z}_i \in [1..(2 \cdot \text{length}(s))]$$

for every $1 \leq i \leq n$. The self-avoidingness is just $(\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i) \neq (\mathtt{X}_j, \mathtt{Y}_j, \mathtt{Z}_j)$ for $i \neq j$.[2] Next we want to express that the distance between two successive monomers is 1, i.e.

$$\|(\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i) - (\mathtt{X}_{i+1}, \mathtt{Y}_{i+1}, \mathtt{Z}_{i+1})\| = 1$$

Although this is some sort of constraint on the monomer position variables $\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i$ and $\mathtt{X}_{i+1}, \mathtt{Y}_{i+1}, \mathtt{Z}_{i+1}$, this cannot be expressed directly in most constraint programming languages. Hence, we must introduce for every monomer i with $1 \leq i < \text{length}(s)$ three variables $\mathtt{Xdiff}_i$, $\mathtt{Ydiff}_i$ and $\mathtt{Zdiff}_i$. These variables have values 0 or 1. Then we can express the unit-vector distance constraint by

$$\begin{array}{lllll} \mathtt{Xdiff}_i & = & |\mathtt{X}_i - \mathtt{X}_{i+1}| & \quad \mathtt{Zdiff}_i & = & |\mathtt{Z}_i - \mathtt{Z}_{i+1}| \\ \mathtt{Ydiff}_i & = & |\mathtt{Y}_i - \mathtt{Y}_{i+1}| & \quad 1 & = & \mathtt{Xdiff}_i + \mathtt{Ydiff}_i + \mathtt{Zdiff}_i. \end{array}$$

The constraints described above span the space of all possible conformations. I.e., every valuation of $\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i$ satisfying the constraints introduced above is an *admissible* conformation for the sequence s, i.e. a self-avoiding walk of s. Given partial information about $\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i$ (expressed by additional constraints as introduced by the search algorithm) we call a conformation c *compatible* with these constraints on $\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i$ if c is admissible and c satisfies the additional constraints.

But in order to use constraint optimization, we have to encode the energy function. For HP-type models, the energy function can be calculated if we know for every pair of monomers (i, j) whether i and j form a contact. i and j form a *contact* in a conformation c, if $j \notin \{i-1, i, i+1\}$ and

$$||c(i) - c(j)|| = 1.$$

For this purpose we introduce for every pair (i, j) of monomers with $i + 1 < j$ a variable $\mathtt{Contact}_{i,j}$. $\mathtt{Contact}_{i,j}$ is 1 if i and j have a contact in every conformation which is compatible with the valuations of $\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i$, and 0 otherwise. Then we can express this property in constraint programming as follows:

$$\begin{array}{rcll} \mathtt{Xdiff}_{i,j} & = & |\mathtt{X}_i - \mathtt{X}_j| & \quad \mathtt{Zdiff}_{i,j} = |\mathtt{Z}_i - \mathtt{Z}_j| \\ \mathtt{Ydiff}_{i,j} & = & |\mathtt{Y}_i - \mathtt{Y}_j| & \quad \mathtt{Contact}_{i,j} \in \{0, 1\} \\ (\mathtt{Contact}_{i,j} = 1) & \Leftrightarrow & (\mathtt{Xdiff}_i + \mathtt{Ydiff}_i + \mathtt{Zdiff}_i = 1) & \end{array} \quad (2)$$

where $\mathtt{Xdiff}_{i,j}$...$\mathtt{Zdiff}_{i,j}$ are new variables. The constraint (2) is called a reified constraint, and can be directly encode in Oz [26].

[2] This cannot be directly encoded in Oz [26], but we reduce these constraints to difference constraints on integers.

Using the variables $\texttt{Contact}_{i,j}$, we can now easily encode the energy function for HP-type models. This means that we can now define a variable `Energy` which is subject to constraint optimization. For the HP-model, we get the constraint

$$\texttt{Energy} = \sum_{\texttt{i+1<j} \wedge \texttt{s(i)=H} \wedge \texttt{s(j)=H}} -\texttt{Contact}_{\texttt{i,j}}.$$

For the HPNX-model, the corresponding constraint can be generated analogously using the energy matrix given in (1).

Thus, we have encoded self-avoiding walks together with a variable `Energy`. Now we can describe the search procedure, which is a combination of generate-and-constraint and branch-and-bound. In a generate step, a undetermined variable *var* out of the set of variables $\{\texttt{X}_i, \texttt{Y}_i, \texttt{Z}_i \mid 1 \leq i \leq n\}$ is selected (according to some strategy). A variable is *determined* if its associated domain consists of only one value, and *undetermined* otherwise. Then, a value *val* out of the associated domain is selected and the variable is set to this value in the first branch (i.e., the constraint $var = val$ is inserted), and the search algorithm is called recursively. In the second branch, which is visited after the first branch is completed, the constraint $var \neq val$ is added.

Each insertion of a constraint leads through constraint propagation to narrowing of some (or many) domains of variables or even to failure, which both prune the search tree by removing inconsistent alternatives. Thus the search is done by alternating constraint propagation and branching with constraint insertion. The generate-and-constraint steps are iterated until all variables are determined (which implies, that a valid conformation is found). If we have found a valid conformation c, then the constraints will guarantee that `Energy` is determined. Let E_c be associated value of `Energy`. Then the additional constraint

$$\texttt{Energy} \;<\; E_c \tag{3}$$

is added, and the search is continued in order to find the next best conformation, which must have a smaller energy than the previous ones due to the constraint (3). This implies that the algorithm finally finds a conformation with minimal energy.

At every node n of the search tree, we call the set of constraints introduced by the search algorithm so far the *configuration* at node n. Every conformation that is found below node n in the search tree must be compatible with the configuration at n, and vice versa. A *bounding function for* `Energy` is a function that takes a configuration of some node n, and yields some value E, where every conformation compatible with the configuration of n has an energy greater than E.

2.3 Redundant Constraints

Clearly, the above described constraint problem generated from a sequence s is not sufficient to yield an efficient implementation. For efficiency, one needs 1.) effective bounding functions; 2.) the ability for implementing a search strategy that tends to

enumerate low energy conformations first. This will be achieved by using redundant variables and constraints (i.e., constraints, which can be removed without losing correctness, but allow the above described pruning). The extension needed for the HPNX-model will be described informally in the next section. A more formal presentation of the redundant constraints for for the HP-model are given in [3], and for the HPNX-model in [4]. We will now describe how constraint programming can be used for adding redundant constraints easily.

When searching for a solution, it is a good search strategy not to fix monomer positions monomer per monomer, but to determine only the value of the x-coordinate of the monomers first. This is the same as determining distribution of monomers to layers (i.e., to planes orthogonal to the x-axis). Now given such a distribution of monomers to layers, then we can apply a bounding function on the Surface of the H-core given the number of even and odd monomers in every layer. This can be achieved by adding two simple constraints. For every layer defined by the equation $x = c$ and every monomer i, we introduce a Boolean variables $\texttt{Elem}_i^{x=c}$, which is defined by the so-called reified constraint

$$\texttt{Elem}_i^{x=c} \Leftrightarrow (\texttt{X}_i = c)$$

Now we can count the number $\texttt{E}_{x=c}\texttt{.seh}$ (resp. $\texttt{E}_{x=c}\texttt{.soh}$ of even (resp. odd) H-monomers in layer $x = c$ simply be the constraint

$$\texttt{E}_{x=c}\texttt{.seh} = \sum_{i \text{ even and } s_i = H} \texttt{Elem}_i^{x=c}$$

(resp. $\texttt{E}_{x=c}\texttt{.soh} = \sum_{i \text{ odd and } s_i = H} \texttt{Elem}_i^{x=c}$). The constraint machinery will guarantee that $\texttt{E}_{x=c}\texttt{.seh}$ will have proper bounds at every search step. Since all constraints work in two directions, we can apply the bounding function to the values of the set of variables $\{\texttt{E}_{x=c}\texttt{.seh}\}$ and $\{\texttt{E}_{x=c}\texttt{.soh}\}$, thus restricting the possible values of the set of variables $\{\texttt{X}_i\}$.

We will give another example for an redundant constraint in the HPNX-model. As we will describe in the next section, it is important to calculated during the search which types of monomers can be placed on a specific position $\vec{p} = (p_x, p_y, p_z)$. This is captured by introducing for every position $\vec{p}$ variables $\texttt{Htype}_{\vec{p}}$, $\texttt{Ptype}_{\vec{p}}$, $\texttt{Ntype}_{\vec{p}}$ and $\texttt{Xtype}_{\vec{p}}$, stating whether the $\vec{p}$ is occupied by an H-,P-,N- or X-monomer[3]. These variables can be defined easily by reified constraints. Thus, $\texttt{Ptype}_{\vec{p}}$ is defined by

$$\texttt{Ptype}_{\vec{p}} \leftrightarrow [(\texttt{X}_i = p_x) \wedge (\texttt{Y}_i = p_y) \wedge (\texttt{Z}_i = p_z)],$$

which can be stated directly this way in constraint programming. Now given the variables $\texttt{Ptype}_{\vec{p}}$ and $\texttt{Ntype}_{\vec{p}}$, then one can apply a bounding function on the energy contribution of the P- and N-monomers as described in the next section. Since the constraints work in both direction, adding the constraint $\texttt{Ptype}_{\vec{p}}$ will immediately exclude all non P-monomers from this position. Thus, we are free in the search

[3] in the case of $\texttt{Xtype}_{\vec{p}}$, we additionally subsume the case that $\vec{p}$ is not occupied at all

strategy whether we will enumerate the monomer positions directly, or whether it is better to enumerate monomer types before. The first strategy is better if there are many P and N-monomers, the second is better if there are not too much such monomers, and we have already found conformations with many PN-contacts.

To summarize, constraint programming gives us the freedom to introduce easily redundant constraints. Furthermore, it allows to optimize the search strategy since all constraints work in both directions.

2.4 HPNX Extensions

As a benefit of our constraint programming approach it is possible to extend the HP algorithm to find the native structure of HPNX proteins.

Since one can see the HP model as embedded into HPNX, resp. HPNX as an extension to HP, a first naive approach to do such an extension is as follows. First, search all HP-optimal conformations of an HPNX sequence, i.e. the conformations that have maximal H-H-contacts. Then, second, find in the set of the HP-optimal conformations the ones with optimal HPNX-energy. This approach is certainly inefficient, since one has a lot of search steps because of the high degeneracy of the HP-model. But further it yields only those native HPNX-conformations that are also optimal in HP, but this is not necessarily true (although this didn't occur in our test set).

Our approach starts by updating the energy constraint. Now we get

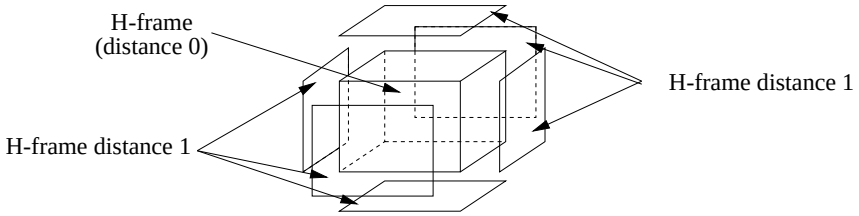
$$\texttt{Energy} \;=\; -4 \cdot \mathit{HH\_Contacts} - \mathit{PN\_Contacts} + \mathit{PP\_Contacts} + \mathit{NN\_Contacts},$$

where HH_Contacts, PP_Contacts resp. NN_Contacts is the number of contacts between H, P resp. N monomers and PN_Contacts the number of contacts between P and N monomers.

To get an efficient implementation, we additionally need a good lower bound on the PN-energy, i.e., $-\mathit{PN\_Contacts} + \mathit{PP\_Contacts} + \mathit{NN\_Contacts}$. The details of this lower bound are described in [4]. Basically, we need to calculate during the search for every position which types of monomers can be placed at this positions (which is called the type of the position). If the position types are fixed, then one can read off the PN-energy from the distribution of the positions types. One has just to count the number of contacts between the corresponding position types, and to subtract the number of bonds of the corresponding types. If only partial information is given about the position types, then one can get bounds form this partial information.

Another concept that is used for efficiency is the concept of an compartment. Fix an H-frame f. We define a *compartment C with H-frame distance d* as a maximal, connected set of points, where all points have the same H-frame distance d. Note, that according to our definition there is a single compartment with H-frame distance 0, which is just the H-frame. Higher order compartments are placed around the H-frame as planes, lines and points. The compartments with H-frame distance 0 and

1 are as follows:



Now this concept helps us pruning the search tree in two ways. First, not every polar monomer i can be member of any compartment C. Instead, there is a restriction which depends on the H-frame distance of C and the position of i in the sequence. Second, the compartments restrict the possible assignment of types to positions.

3 Results

	Sequences and Sample Conformations
S1hp	HPPPPHHHHPPHPHPHHHPHPPHHPPH
S1	HXXNNHHHHXPHXHXHHHNHPPHHXPH
	RDLLUBRUULDLFFRBRFDDLULDBU
S2hp	HPPPHHHHPHPHHPPPHPHHPHPPPHP
S2	HXNNHHHHXHXHHNXNHXHHNHPPXHP
	RBULDLFLUBRFULFDDRRULURBDR
S3hp	HPHHPPHHPPHHHHPPPHPPPHHHPPH
S3	HPHHNXHHPNHHHHXXXHXPXHHHPXH
	LFRURDDDBULFLBBURUFRDBDDLU
S4hp	HHPHHPHHPHHHHHHPPHHHHHPPHHHHHHHH
S4	HHXHHPHHXHHHHHHPNHHHHHPNHHHHHHHH
	LLFDBRDLFRRBUFRULFDDLFUBBUFLDD
S5hp	PHPPHPPHPPHPPHPPHPPHPPHPPHPPHPPHPPHP
S5	XHXNHXXHNPHXXHPXHXXHXNHPXHNXHPXHXPHX
	LFUBUBDBLFUFDFDBDLUFUBUBDBDFDRUBRFR

Table 1: Test sequences. We have listed the test sequences S1–S4, together with the corresponding HP-sequence and an optimal conformation (using absolute moves, where L means 'left', U means 'up', 'F' means 'forward' and so on).

We investigate a set of test sequences as shown in Table 1. Here, we have grouped the sequences, such that for every group i there is a HP-sequence S$i$hp, from which the other sequences are generated by replacing P monomers by P, N and X monomers. The splitting was done at random (where P-monomers are split into P-,N- and X-monomers with probabilities 0.25, 0.25 and 0.50). We will call the S$i$hp the generating HP-sequence of the HPNX-sequences in i.

Sequence	Search Steps Best HPNX	Search Steps All HP	HPNX Degeneracy	HP Degeneracy
S1	14402	167662	61	37244
S2	733	2998	4	297
S3	411	155693	195	25554
S4	46	11036	1023	1114
S5	1629	55086	16	3528

Table 2: Results. We have compared the number of search steps for finding the optimal HPNX-sequence with the number of search steps to find all HP-optimal sequences. Furthermore, we have compared the degeneracy in the HPNX- and HP-model for some sample sequences as found by our algorithm.

The algorithm finds the native structure of all sequences listed in Table 1. For the HP-sequences, results from Yue and Dill [29] have been shown to be reproducible with our implementation in an earlier paper [3]. For the HPNX-model, to our knowledge, there is no other algorithm that allows to find provably optimal conformations.

Note that there is a difference between finding the native structure, and proving that the best found structure is really the optimal one (which requires that the complete search space has been investigated). Hence, we display in Table 2 the search steps needed to find the native conformation (# steps find), and the number of steps needed to show that the best found conformation is really optimal (# steps prove).

This proves that, the implementation of the algorithm improved the procedure by more than one order of magnitude.

Furthermore, we have compared the degeneracy of the HPNX-sequences with the corresponding HP-sequences in Table 2. One can find that the degeneracy is strongly reduced in the HPNX-model.

From theoretical considerations it is to be expected that larger alphabets imply smaller degeneracies since the space of pre-images for the sequence space to shape space map is larger. This has also been conjectured from the random energy model [16] and from landscape computations with a heuristic approximation algorithm for lattice models [23].

Here we explicitly show such a case for a set of particular HPNX sequences that were derived from their HP sequences where the H were left at their place and the P-residues randomly substituted by either P, N, or X.

In Fig. 2, we show some selected examples of S2. All four ground state configurations of the HPNX sequence are given. It can be seen that they have a very similar shape, pairwisely differing only by two moves or the combination of both moves. The four randomly selected examples (out of 297 possible ones) of degenerate ground state structures from the corresponding HP sequence (see Fig. 3) show a much higher structural variation with no obvious overlap.

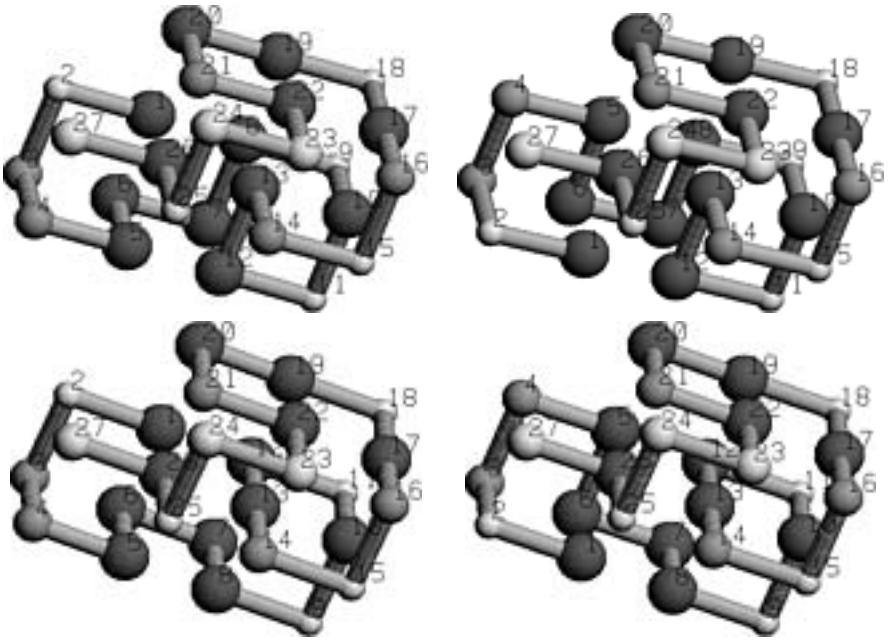


Figure 2: The 4 optimal conformations for sequence S2. Big beads are H-monomers, middle-sized ones are P or Ns, and small ones are X.

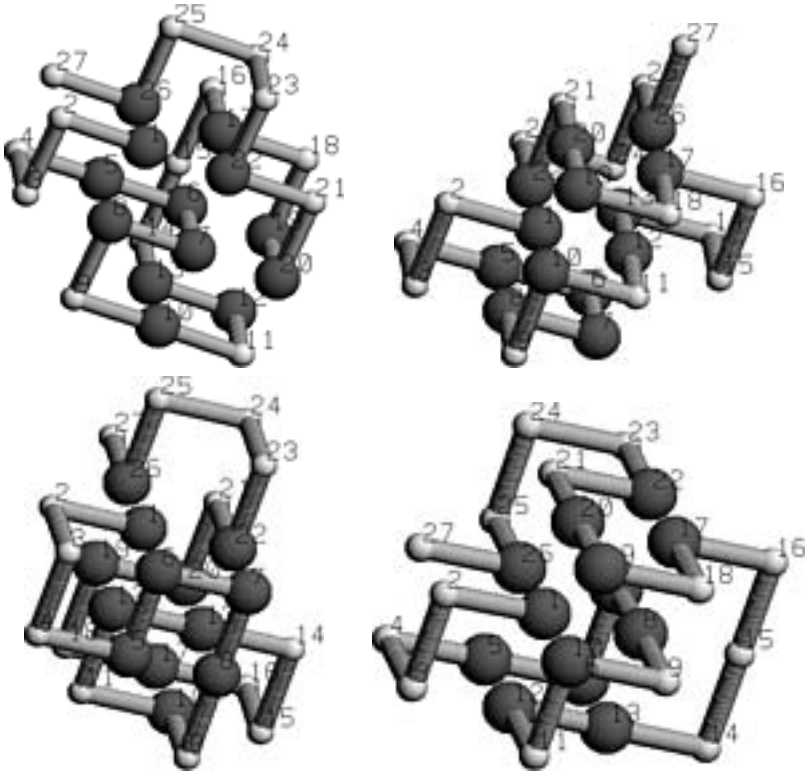


Figure 3: 4 HP-optimal conformations for S2hp (out of 297)

4 Discussion

Here we reported on the application of a new technique that is valuable to test theories and hypothesis about the effect of using CP for the structure prediction of lattice proteins when using extended alphabets.

1. First of all we have explicitly shown that CP is a most powerful technique to cope with optimization problems of considerable size and complexity. Therefore, the pruning is definitely possible for higher dimensional lattices as well as long a regular lattice can be entirely represented by integers. We expect further improvements by new development of the platform and the language Oz.

2. We have shown that higher order alphabets can be handled and reduce the degeneracy of solution structures. This was not achieved at the price of confined solution space. To show the correspondence to HP sequences we used a very similar and simple potential. It is clear that many of the benefits arise from modeling the energy constraints as an ad-hoc assumption with a strong over-representation of the HH values. In fact, any alphabet, even more realistic ones as presented in [6], and larger one can be used. However, several examples from literature have shown that reduced alphabets can in many ways be sufficient to find a good solution or at least to narrow down a search to a small number of potential solutions in problems such as structure representation and sequence alignment. This complies with the fact that a relatively small number of residues with a characteristic polar non-polar pattern is sufficient to construct a real protein [22, 17].

3. Because all solution structures were non-MCS our results reconfirm (for earlier results from the HP model see also [30, 7]) that confining search space to a restricted shape space is a simplification which is not always justified even though the average attraction force in the potential is very high. Furthermore, it is intuitively clear that using MCS alone will reduce the shape space drastically by many orders of magnitude. As a result, since sequence space remains the same, reduced degeneracy (which itself is an arguable disadvantage, see [9] for discussion) must at least partly be attributed to usage of the smaller search space. Therefore, while general conclusions that are drawn on ensemble properties from models confined to MCS [15, 20] remain unaffected, we think that the simplification of using MCS is an equally drastic simplification as using reduced alphabets or simpler lattices. Consequently, systems as used in [28, 15, 20, 1] should be considered as models with a *different* simplification and not superior just because one simplification (reduced alphabet) is replaced by another (reduced solution space). These problems are circumvent by our method.

Applying more fine grained lattices has proved useful in kinetic folding simulations [25]. It can also be useful for our approach, since the formulation of self-

avoiding chains, position types, and surface extends easily to other lattices. It can be expected that even part of the search strategy can be applied in the case of a different lattice. What is required is an equivalent to the concept of a H-frame, and the definition of corresponding bounding functions.

The extension of our approach to more fine grained lattices we deem instructive for development of techniques which are much closer to solving realistic problems rationally than most existing ones.

Acknowledgment We would like to thank Prof. Peter Clote, who got us interested in bioinformatics and inspired this research. RB would like to thank Prof. Martin Karplus for helpful discussions on the topic of lattice models, and for motivating him to apply constraint programming techniques to lattice protein folding. EBB acknowledges financial support by the Klaus Tschira Stiftung (KTS)

References

[1] V. I. Abkevich, A. M. Gutin, and E. I. Shakhnovich. Impact of local and nonlocal interactions on thermodynamics and kinetics of protein folding. *Journal of Molecular Biology*, 252:460–471, 1995.

[2] V.I. Abkevich, A.M. Gutin, and E.I. Shakhnovich. Computer simulations of prebiotic evolution. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, and Teri E. Klein, editors, *PSB'97*, pages 27–38, 1997.

[3] Rolf Backofen. Constraint techniques for solving the protein structure prediction problem. In *Proceedings of* 4^{th} *International Conference on Principle and Practive of Constraint Programming (CP'98)*, 1998.

[4] Rolf Backofen and Sebastian Will. A branch-and-bound constraint optimization approach to the hpnx structure prediction problem. 1998. submitted.

[5] B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (HP) modell is NP-complete. In *Proc. of the RECOMB'98*, pages 30–39, 1998.

[6] Erich Bornberg-Bauer. Chain growth algorithms for HP-type lattice proteins. In *Proc. of the* 1^{st} *Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 47 – 55. ACM Press, 1997.

[7] Erich Bornberg-Bauer. How are model protein structures distributed in sequence space ? *Biophys. J.*, 73:2393–2403, 1997.

[8] Hue Sun Chan. Kinetics of protein folding. *Nature*, 373:664 – 665, 1995.

[9] Hue Sun Chan and Ken A. Dill. Comparing folding codes for proteins and polymers. *Proteins*, 24:335 – 344, 1996.

[10] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. In *Proc. of STOC*, 1998. To appear. Short version in *Proc. of RECOMB'98*, pages 61–62.

[11] Gordon M. Crippen. Prediction of protein folding from amino acid sequence over discrete conformation spaces. *Biochemistry*, 30:4232 – 4237, 1991.

[12] 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 of simple exact models. *Protein Science*, 4:561–602, 1995.

[13] Aaron R. Dinner, Andreaj Šali, and Martin Karplus. The folding mechanism of larger model proteins: Role of native structure. *Proc. Natl. Acad. Sci. USA*, 93:8356–8361, 1996.

[14] S. Govindarajan and R. A. Goldstein. Evolution of model proteins on a foldability landscape. *Proteins*, 29(4):461–466, 1997.

[15] S. Govindarajan and R. A. Goldstein. The foldability landscape of model proteins. *Biopolymers*, 42(4):427–438, 1997.

[16] A. M. Gutin and E. I. Shakhnovich. Ground state of random copolymers and the discrete random energy model. *J. Chem. Phys.*, 98:8174 – 8177, 1993.

[17] Satwik Kamtekar, Jarad M. Schiffer, Huayu Xiong, Jennifer M. Babik, and Michael H. Hecht. Protein design by binary patterning of polar and nonpolar amino acids. *Science*, 262:1680 – 1685, 1993.

[18] Kit Fun Lau and Ken A. Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22:3986 – 3997, 1989.

[19] Kit Fun Lau and Ken A. Dill. Theory for protein mutability and biogenesis. *Proc. Natl. Acad. Sci. USA*, 87:638 – 642, 1990.

[20] Hao Li, Robert Helling, Chao Tnag, and Ned Wingreen. Emergence of preferred structures in a simple model of protein folding. *Science*, 273:666–669, 1996.

[21] Angel R. Ortiz, Andrzej Kolinski, and Jeffrey Skolnick. Combined multiple sequence reduced protein model approach to predict the tertiary structure of small proteins. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, and Teri E. Klein, editors, *PSB'98*, volume 3, pages 375–386, 1998.

[22] J.F. Reidhaar-Olson and R.T. Sauer. Combinatorial casette mutagenesis as a probe of the information content of protein. *Science*, 241:53 – 57, 1988.

[23] Alexander Renner and Erich Bornberg-Bauer. Exploring the fitness landscapes of lattice proteins. In Russ Altman, Keith Dunker, Lawrence Hunter, and Terri Klein, editors, *Proceedings of the 1997 Pacific Symposium on Biocomputing*, pages 361 – 373. World Scientific, London, 1997.

[24] Indira Shrivastava, Saraswathi Vishveshwara, Marek Cieplak, Amos Maritan, and Jayanth R. Banavar. Lattice model for rapidly folding protein-like heteropolymers. *Proc. Natl. Acad. Sci., USA*, 92:9206–9209, 1995.

[25] Jeffrey Skolnick and Andrzej Kolinski. Simulations of the folding of a globular protein. *Science*, 250:1121 – 1125, 1990.

[26] Gert Smolka. The Oz programming model. In Jan van Leeuwen, editor, *Computer Science Today*, Lecture Notes in Computer Science, vol. 1000, pages 324–343. Springer-Verlag, Berlin, 1995.

[27] Ron Unger and John Moult. Local interactions dominate folding in a simple protein model. *Journal of Molecular Biology*, 259:988–994, 1996.

[28] A. Šali, E. Shakhnovich, and M. Karplus. Kinetics of protein folding. *Journal of Molecular Biology*, 235:1614–1636, 1994.

[29] Kaizhi Yue and Ken A. Dill. Sequence-structure relationships in proteins and copolymers. *Physical Review E*, 48(3):2267–2278, September 1993.

[30] Kaizhi Yue, M. Fiebig, Poaul D. Thomas, Hue Sun Chan, Eugene I. Shakhnovich, and Ken A. Dill. A test of lattice protein folding algorithms. *Proc. Natl. Acad. Sci. USA*, 92:325 – 329, 1995.