Constraint Techniques for Solving the Protein Structure Prediction Problem

Rolf Backofen

Institut für Informatik/LMU München
Oettingenstraße 67, D-80538 München
backofen@informatik.uni-muenchen.de

Abstract. The protein structure prediction problem is one of the most (if not *the most*) important problem in computational biology. This problem consists of finding the conformation of a protein (i.e., a sequence of amino-acids) with minimal energy. Because of the complexity of this problem, simplified models like Dill's HP-lattice model [12] have become a major tool for investigating general properties of protein folding. Even for this simplified model, the structure prediction problem has been shown to be NP-complete [3, 5].

We describe a constraint formulation of the HP-model structure prediction problem, present the basic constraints and search strategy. We then introduce a novel, general technique for excluding geometrical symmetries in constraint programming. To our knowledge, this is the first general and declarative technique for excluding symmetries in constraint programming that can be added to an existing implementation. Finally, we describe a new lower bound on the energy of an HP-protein. Both techniques yield an efficient pruning of the search tree.

1 Introduction

The protein structure prediction problem is specified as follows: Given a protein by its sequence of amino acids, what is its native structure? Many results in the past have shown the problem to be NP-hard. But the situation is even worse, since one does not know the general principles why natural proteins fold into a native structure. E.g., these principles are interesting if one wants to design artificial proteins (for drug design). For the time being, one problem there is that artificial proteins usually don't have a native structure.

To attack this problem, simplified models have been introduced, which became a major tool for investigating general properties of protein folding. An important class of simplified models are the so-called lattice models. The simplest used lattice is the cubic lattice, where every conformation of a lattice protein is a self-avoiding walk in $\mathbb{Z}^3$. A discussion of lattice proteins can be found in [6]. There is a bunch of groups working with lattice proteins. Examples of how lattice proteins can be used for predicting the native structure or for investigating principles of protein folding are [17, 1, 8, 16, 11, 9, 2, 13].

(a)

	H	P
H	-1	0
P	0	0

(b)

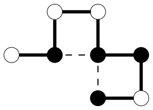


Fig. 1. Energy matrix and sample conformation for the HP-model

An important representative of lattice models is the HP-model, which has been introduced by [12]. In this model, the 20 letter alphabet of amino acids 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, and the euclidian distance of the positions is 1. A conformation with *minimal energy* (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 [3, 5].

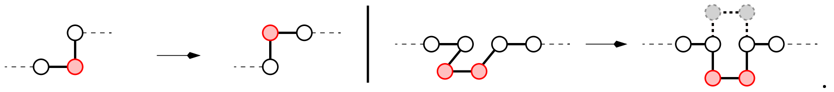
A sample conformation for the sequence PHPPHHPH in the two-dimensional lattice with energy -2 is shown in Figure 1(b). The white beads represent P, the black ones H monomers. The two contacts are indicated via dashed lines.

An example of the use of lattice models is the work by Šali, Shakhnovich and Karplus [17].[1] They investigate under which conditions a protein folds into its native structure by performing the following computer experiment:

1.) generate 200 random sequences of length 27.

2.) find the minimal structures on the $3 \times 3 \times 3$-cube. The reason for using a sequence length of 27 is that the $3 \times 3 \times 3$-cube has exactly 27 position.[2]

3.) simulate protein folding on the lattice model using a Monte Carlo method with Metropolis criteria. The Monte Carlo method is as follows. Initially, a random conformation of the sequence is generated. Starting from this initial conformation, the algorithm performs so-called Monte Carlo steps in order to search for the minimal conformation. A single Monte Carlo step consists of the following operations: First, a local move is selected at random until a move is found that produces a valid conformation (i.e., a self-avoiding conformation). Two examples of allowed moves are



Here, the positions of the shaded monomers are changed. Second, the resulting conformation is evaluated according to the Metropolis criterion. If the energy of the result is lower than the energy of the previous one, then the conformation is always accepted. Otherwise, the conformation is accepted by random, where the probability depends on the energy difference.

[1] The same lattice model is used by several other people, e.g., [1, 16, 2, 9].

[2] In a later paper [8], the authors considered proteins of length 125.

Now a protein folds in that framework, if the Monte Carlo method finds its native conformation (by performing 50 000 000 Monte Carlo steps). The authors have found that a protein folds if there is a energy gap between the native structure and the energy of the next minimal structure.

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 as required by Step 2. For the energy function used by [17], there is no *exact* algorithm for finding the minimal structure. To be computational feasible, they have restricted in [17] the search for the native structure on the $3 \times 3 \times 3$-cube as indicated in Step 2.

Previous Work In the literature, several algorithms were proposed for the HP-model. E.g., there are heuristic approaches such as the hydrophobic zipper [7], the genetic algorithm by Unger and Moult [15] and the chain growth algorithm by Bornberg-Bauer [4]. Another example is an approximation algorithm as described in Hart and Istrail [10], which produces a conformation, whose energy is known to be at least $\frac{3}{8}$ of the optimal energy, in linear time. And there is one exact algorithm, namely the CHCC of Yue and Dill [18], which finds all optimal conformations. There are two differences between the CHCC-algorithm and ours. First, the motivation for development of CHCC was to find *all* minimal conformations in the HP-model, whereas we are only interested in finding the minimal energy. Second, we want to provide a declarative formulation of the problem that can be used for other models as well (currently, we are working on an extension of the HP-model). The CHCC algorithm is designed in a way that is only suited for the HP-model.

Contributions and Plan of the Paper We have transformed the protein structure prediction problem to a constraint minimisation problem with finite domain variables, Boolean variables, and reified constraints. We have then implemented this constraint problem using the language Oz [14]. The main problem we where faced with was the existence of 47 geometrical symmetries. One possible way for excluding symmetries is to use an appropriate modeling. Although this results in an efficient implementation in general, this approach has some drawbacks. Despite the fact that one often does not find a modeling that excludes the symmetries (as in our case), this approach is inflexible. Usually, such a model cannot be extended without doing a complete re-modeling.

For this reason, we have searched for a declarative way of excluding symmetries. In our approach, we consider binary branching search trees. The symmetries are excluded by adding at the right branch (which is visited after the left branch) constraints which enforce the right branch to exclude all solution for which a symmetric solution has been found in the left branch. These exclusion constraints are defined by just using general properties of the symmetries considered. There are several advantages. First, it is a general method that can be used with any kind of symmetries that can be defined using constraint expressions. Second, it can be added to an existing implementation, since this technique is applied on the level of the search tree, and uses existing constraint expressions. And third, it does not impose any restrictions on the search strategy. To our knowledge, there is no existing method for excluding symmetries declaratively.

Another way to prune the search tree was the use of a new lower bound on the surface of all H-monomers given their distribution to planes described by the equation $x = c$. This results in an upper bound on the number of contacts. The lower bound on the surface uses a property of lattice models, namely that for any sequence s and any conformation of s in $\mathbb{Z}^3$, two monomers $1 \leq i, j \leq \mathrm{length}(s)$ can form a contact iff $|i - j| > 1$, and i is even and j is odd, or vice versa.

In Section 2.1, we introduce the basic definitions for the structure prediction problem. In Section 2.2, we introduce the constraint minimisation problem modeling the structure prediction problem and describe the search strategy. We then introduce in Section 2.3 the technique for excluding symmetries in a declarative way, and apply the introduced technique to our lattice problem. In the following Section 2.4, we explain the new lower bound on the surface. Finally, in Section 3, we present results for some HP-sequences taken from the literature, show search times and number of search steps with and without symmetry exclusion.

2 Constraint Formulation

2.1 Basic Definitions

A sequence is an element in $\{H, P\}^*$. With s_i we denote the i^{th} element of a sequence s. We say that a monomer with number i in s is even (resp. odd) if i is even (resp. odd). A conformation c of a sequence s is a function $c : [1..|s|] \rightarrow \mathbb{Z}^3$ such that

1. $\forall 1 \leq i < |s| : ||c(i) - c(i+1)|| = 1$ (where $|| \cdot ||$ is the euclidic norm on $\mathbb{Z}^3$)
2. and $\forall i \neq j : c(i) \neq c(j)$.

Given a conformation c of a sequence s, the number of contacts $\mathrm{Contact}_s(c)$ in c is defined as the number of pairs (i, j) with $i + 1 < j$ such that

$$s_i = H \wedge s_j = H \wedge ||c(i) - c(j)|| = 1.$$

The energy of c is just $-\mathrm{Contact}_s(c)$. With $\boldsymbol{e}_x$, $\boldsymbol{e}_y$ and $\boldsymbol{e}_z$ we denote $(1, 0, 0)$, $(0, 1, 0)$ or $(0, 0, 1)$, respectively. We say that two points $\boldsymbol{p}, \boldsymbol{p}' \in \mathbb{Z}^3$ are *neighbors* if $||\boldsymbol{p} - \boldsymbol{p}'|| = 1$. Then the *surface* $\mathrm{Surf}_s(c)$ is defined as the number of pairs of neighbor positions, where the first position is occupied by an H-monomer, but the second not. I.e.,

$$\mathrm{Surf}_s(c) = \left|\left\{ (c(i), \boldsymbol{p}) \middle| s_i = H \ \wedge \ ||\boldsymbol{p} - c(i)|| = 1 \wedge \forall j : (s_j = H \Rightarrow c(j) \neq \boldsymbol{p}) \ \right\}\right|$$

Now Yue and Dill [18] made the observation that there is a simple linear equation relating surface and energy. This equation uses the fact that every monomer has 6 neighbors, each of which is in any conformation either filled with either an H-monomer, a P-monomer, or left free. Let n_H^s be the number of H-monomers in s. Then we have for every conformation c that

$$6 \cdot n_H^s = 2 \cdot [\mathrm{Contact}_s(c) + \mathrm{HHBonds}(s)] + \mathrm{Surf}_s(c), \tag{1}$$

where HHBonds(s) is the number of bonds between H-monomers (i.e., the number of H-monomers whose successor in s is also a H-monomer). Since HHBonds(s) is constant for all conformations c of s, this implies that minimizing the surface is the same as maximizing the number of contacts.

In a later section, we will consider a lower bound on the surface given partial knowledge about a conformation c. Given the above, the lower bound on the surface yields an upper bound on the number of contacts (which generates in fact a lower bound on the energy since the energy is defined as $-\mathrm{Contact}_s(c)$).

Given a conformation, the *frame* of the conformation is the minimal rectangular box that contains all H-monomers of the sequence. Given a vector $\boldsymbol{p}$, we denote with $(\boldsymbol{p})_\mathrm{x}$, $(\boldsymbol{p})_\mathrm{y}$ and $(\boldsymbol{p})_\mathrm{z}$ the x-,y- and z-coordinate of $\boldsymbol{p}$, respectively. The *dimensions* (fr_x, fr_y, fr_z) of the frame are the numbers of monomers that can be placed in x-, y- and z-direction within the frame. E.g., we have

$$fr_x = \max\{|(c(i) - c(j))_\mathrm{x}| \mid 1 \leq i, j \leq \mathrm{length}(s) \wedge s_i = H \wedge s_j = H\} + 1.$$

2.2 Constraints and Search Strategy

A frame is uniquely determined by its dimension and its starting point. Yue and Dill [18] provided a method to calculate a lower bound on the surface when all H-monomers are packed within a specific frame. Thus, there are usually a few frames to be searched through to find the optimal conformation, since often bigger frames have a higher lower bound for the surface than an optimal conformation found in a smaller frame. For all examples in [18], there is even only one frame that has to be searched through. Note that also some of the P-monomers must be included within this frame, namely those P-monomers whose left and right neighbors in chain are H-monomers. The reason is just that one cannot include the surrounding H-monomers into the core without also including the middle P-monomer. These P-monomers are called *P-singlets* in [18]. A position $p \in \mathbb{Z}^3$ is a *caveat in a conformation c of s* if p is contained in the hull (over $\mathbb{Z}^3$) of the set of positions occupied by H-monomers in c

Our constraint problem consists of finite domain variables. We use also Boolean constraint and reified constraints. With reified constraints we mean a constraint x =: (ϕ), where ϕ is a finite domain constraint. x is a Boolean variable which is 1 if the constraint store entails ϕ, and 0 if the constraint store disentails ϕ. A constraint store entails a constraint ϕ if every valuation that makes the constraint store valid also makes ϕ valid. We use also entailment constraints of the form $\phi \rightarrow \psi$, which are interpreted as follows. If a constraint store entails ϕ, then ψ is added to the constraint store. We have implemented the problem using the language Oz [14], which supports finite domain variables, Boolean constraints, reified constraints, entailment constraints and a programmable search module. The latter was used for the implementation of the symmetry exclusion.

Caveats	Boolean; is 0 if the conformation contains no caveats
Frx, Fry, Frz	dimension of the frame
$\texttt{X}_i$, $\texttt{Y}_i$, $\texttt{Z}_i$	x-,y-, and z-coordinate of the i^{th} monomer
$\texttt{E}_j\texttt{.seh}$, $\texttt{E}_j\texttt{.soh}$	number of even and odd H-monomers of the j^{th} x-plane (or x-layer) in the frame, respectively (where $1 \leq j \leq \texttt{Frx}$);
$\texttt{Elem}_j^i$	membership of H-monomer i in the j^{th} x-layer
$\texttt{P}_k\texttt{.ctp}$	type of the k^{th} position of the frame (where $1 \leq k \leq \texttt{Frx} \cdot \texttt{Fry} \cdot \texttt{Frz}$); the core type $\texttt{P}_k\texttt{.ctp}$ of the k^{th} position is either 1, if it is occupied by an H-monomer, and 0 otherwise
$\texttt{O}_i^k$	for every position k of the frame and every monomer i; $\texttt{O}_i^k$ has boolean value (i.e., 0 or 1), and is 1 iff monomer i occupies the k^{th} position of the frame.
$\texttt{Surf}_k^l$	surface contribution between neighbour positions k and l under the condition, that k is occupied by an H-monomer. Thus, k is in the frame, and l is in the frame or within distance 1 from the frame
Surface	complete surface of the conformation

Fig. 2. The variables and their description

Given a specific sequence s, the main variables of our constraint problem are listed in Figure 2. We use constraint optimization to minimize the variable Surface. There are additional variables and constraints used for pruning the search tree, which we have suppressed for simplicity.

The basic constraints, which describe basic properties of self-avoiding walks, are the following. W.l.o.g., we can assume that we have for every $1 \leq i \leq$ length(s):

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

The self-avoidingness is just $(\texttt{X}_i, \texttt{Y}_i, \texttt{Z}_i) \neq (\texttt{X}_j, \texttt{Y}_j, \texttt{Z}_j)$ for $i \neq j$.[3]

For expressing that the distance between two successive monomers is 1, we introduce for every monomer i with $1 \leq i < \text{length}(s)$ three variables $\texttt{Xdiff}_i$, $\texttt{Ydiff}_i$ and $\texttt{Zdiff}_i$. The value range of these variables is [0..1]. Then we can express the unit-vector distance constraint by

$$\begin{array}{lll} \texttt{Xdiff}_i =: |\texttt{X}_i - \texttt{X}_{i+1}| & & \texttt{Zdiff}_i =: |\texttt{Z}_i - \texttt{Z}_{i+1}| \\ \texttt{Ydiff}_i =: |\texttt{Y}_i - \texttt{Y}_{i+1}| & & 1 =: \texttt{Xdiff}_i + \texttt{Ydiff}_i + \texttt{Zdiff}_i. \end{array}$$

The other constraints are as follows. Clearly, we must have

$$\sum_{j=1}^{\texttt{Frx}} \texttt{E}_j\texttt{.soh} =: |\{i \mid i \text{ odd and } s_i = H\}| \quad \sum_{j=1}^{\texttt{Frx}} \texttt{E}_j\texttt{.seh} =: |\{i \mid i \text{ even and } s_i = H\}|$$

[3] This cannot be directly encoded in Oz [14], but we reduce these constraints to difference constraints on integers.

Then we have for every layer j that $\mathtt{E}_j\mathtt{.soh}+\mathtt{E}_j\mathtt{.seh}+ \leq \mathtt{Fry}\cdot\mathtt{Frz}$. Using reified constraints, $\mathtt{Elem}_j^i$ can be defined by

$$\mathtt{Elem}_j^i =: (\mathtt{X}_i =: j-1 + \text{x-coordinate of starting point of frame}).$$

Then $\mathtt{E}_j\mathtt{.seh} =: \sum_{i \text{ even},\ s_i\, =\, H} \mathtt{Elem}_j^i$, and $\mathtt{E}_j\mathtt{.soh}$ can be defined analogously.

We can state that whenever two monomers i and $i+3$ are in the same layer, then $i+1$ and $i+2$ must also be in one layer due to the condition that we must fold into a lattice conformation. I.e., for every $1 \leq j \leq \mathtt{Frx}$ we have

$$(\mathtt{Elem}_j^i =: 1 \wedge \mathtt{Elem}_j^{i+3} =: 1) \rightarrow \mathtt{X}_{i+1} =: \mathtt{X}_{i+2}$$

Furthermore, there is a special treatment of P-singlets, which may not be buried into the core without forming a caveat. Thus we have for every P-singlet i that

$$(\mathtt{Elem}_j^i =: 1 \wedge \mathtt{Elem}_j^{i+1} =: 0 \wedge \mathtt{Caveats} =: 0) \rightarrow \mathtt{Elem}_j^{i-1} =: 1$$
$$(\mathtt{Elem}_j^i =: 1 \wedge \mathtt{Elem}_j^{i-1} =: 0 \wedge \mathtt{Caveats} =: 0) \rightarrow \mathtt{Elem}_j^{i+1} =: 1.$$

At some stage of the search we have to assign monomers to frame positions. A monomer i is assigned the position k by setting $\mathtt{O}_i^k$ to 1 in one branch (which has just the effect that $\mathtt{Y}_i$ and $\mathtt{Z}_i$ is set to the y- and z-coordinate of the position k), and 0 in the other. Self-avoidingness is achieved by $\mathtt{Sum}[\mathtt{O}_1^k, \ldots, \mathtt{O}_{\mathrm{length}(s)}^k] =<: 1$.

But there are additional constraints which restrict the core type and the monomers that can be placed at some position. Let $\{i_1, \ldots, i_n\}$ be the set of all H-monomers in s. If at some stage no monomer in $\{i_1, \ldots, i_n\}$ can be placed at some position k, then the core type must be 0. This is implemented by

$$\mathtt{P}_k\mathtt{.ctp} =: (\mathtt{Sum}[\mathtt{O}_{i_1}^k, \ldots, \mathtt{O}_{i_n}^k] >: 0).$$

Finally, we have constraints relating core types of positions and surface contributions. Of course, we get $\mathtt{Surface} =: \sum_{\mathrm{k,l}} \mathtt{Surf}_k^l$, where k, l ranges over all neighbor positions. If l is a position outside the frame (i.e., if its x-,y- or z-coordinate is outside the frame), then $\mathtt{Surf}_k^l =: \mathtt{P}_k\mathtt{.ctp}$. Otherwise we have $\mathtt{Surf}_k^l =: (\mathtt{P}_k\mathtt{.ctp} =: 1 \wedge \mathtt{P}_l\mathtt{.ctp} =: 0)$. Now the surface contributions and the $\mathtt{Caveats}$ variable can be related using reified constraints. For every line li in $\mathbb{Z}^3$ parallel to one of the coordinate axis, which intersects with the frame, we define the Boolean variable $\mathtt{Caveat}_{\mathrm{li}}$ by

$$\mathtt{Caveat}_{\mathrm{li}} =: (\textstyle\sum_{k \neq l \text{ on li}} \mathtt{Surf}_k^l >: 2).$$

Then $\mathtt{Caveats} =: (\sum_{\text{lines li}} \mathtt{Caveat}_{\mathrm{li}} >: 1)$.

Our search strategy is as follows. We select the variables according to the following order (from left to right)

$$\mathtt{Caveats} < \begin{array}{c}\mathtt{Frx}\\ \mathtt{Fry}\\ \mathtt{Frz}\end{array} < \begin{array}{c}\mathtt{E}_j\mathtt{.seh}\\ \mathtt{E}_j\mathtt{.soh}\end{array} < \mathtt{Elem}_j^i < \mathtt{O}_i^k < \begin{array}{c}\mathtt{X}_i\\ \mathtt{Y}_i\\ \mathtt{Z}_i\end{array}$$

It is a good strategy to set `Caveats` to 0 in the first branch, since in almost every case there is an optimal conformation without a caveat. The frame dimensions are chosen ordered by surface according to the lower bound given in [18]. After having determined the variables $\mathtt{E}_j\mathtt{.seh}$ and $\mathtt{E}_j\mathtt{.soh}$, we calculate a lower bound on the surface, which will be described in Section 2.4. If all H-monomers and P-singlets are assigned to layers, we search for the positions of these monomers within the frame. Finally, we place the remaining monomers.

2.3 Excluding Geometric Symmetries

We fix a first-order signature Σ including the equality $\doteq$ with a set of variables $\mathcal{V}$. Constraints are literals, and constraint formulae are quantifier-free formulae over Σ. We identify $t \doteq t'$ with $t' \doteq t$. $\mathcal{C}$ denotes the set of all constraints. A set of constraints $C \subseteq \mathcal{C}$ is interpreted as the conjunction of the constraints contained in C, and we will freely mix set notation and conjunction. We fix a standard interpretation $\mathcal{A}$ with domain $\mathcal{D}^{\mathcal{A}}$, which describes our constraint theory. An *assignment* α *in* $\mathcal{A}$ is a partial function $\alpha : \mathcal{V} \to \mathcal{D}^{\mathcal{A}}$. A *propagation operator* $\mathcal{P}$ for $\mathcal{A}$ is a monotone function $\mathcal{P} : \mathcal{C} \mapsto \mathcal{C}$ with $\mathcal{A} \models (C \Leftrightarrow \mathcal{P}(C))$. The propagation operator $\mathcal{P}$ characterises the constraint solver and will be fixed. A constraint set C *determines* a set of variables $\mathcal{X}$ *to an assignment* α iff for all $x \in \mathcal{X}$ there is ground term t such that $\alpha(t) = \alpha(x)$ and $x \doteq t \in C$.

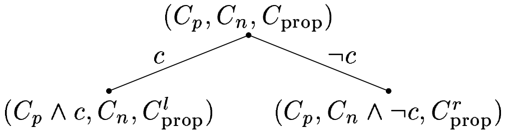
In the following, we assume a fixed constraint set C_{Pr} describing the problem to be solved. Furthermore, our constraint problem has the property, that there is a subset of variables $\mathcal{X} \subseteq \mathcal{V}$ consisting of the monomer position variables $\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i$, whose valuation completely determines the valuation of the other variables. Since we want to define the symmetries on these variables, we define

$$\|C\| = \{\alpha \mid \mathrm{dom}(\alpha) = \mathcal{X} \wedge \mathcal{A}, \alpha \models C\}.$$

where $\mathcal{A}, \alpha \models C$ means that there is a uniquely defined $\alpha' \supseteq \alpha$ total that satisfies C in $\mathcal{A}$. Furthermore, we write $\phi \models \psi$ for entailment, i.e. $\|\phi\| \subseteq \|\psi\|$.

A *symmetry* s *for* C_{Pr} is a bijection $s : \|C_{\mathrm{Pr}}\| \to \|C_{\mathrm{Pr}}\|$. A *symmetry set* $\mathcal{S}$ *for* C_{Pr} is a set of symmetries operating on $\|C_{\mathrm{Pr}}\|$, which is closed under inversion. We denote the identity function on $\|C_{\mathrm{Pr}}\|$ with $\mathrm{id}_{C_{\mathrm{Pr}}}$ (which is a symmetry by definition). Clearly, one can consider the set of all symmetries for C_{Pr} (which even form a group). But in general, we do not want to consider all symmetries, since either there are too many of them, or some of them do not have an intuitive characterisation.

Definition 1 (Search Tree). *Let t be a finite, binary, rooted, ordered tree, whose edges are labelled by literals, and whose nodes are labelled by triples of constraint sets. The tree t is a* search tree for C_{Pr} *if 1.) the root node v_r has the label $(\emptyset, \emptyset, \mathcal{P}(C_{\mathrm{Pr}}))$, and 2.) every binary node has the form*


$(C_p, C_n, C_{\mathrm{prop}})$
c
$\neg c$
$(C_p \wedge c, C_n, C^l_{\mathrm{prop}})$
$(C_p, C_n \wedge \neg c, C^r_{\mathrm{prop}})$

with $C^l_{\text{prop}} \supseteq \mathcal{P}(C_{\text{prop}} \wedge c)$ *and* $C^r_{\text{prop}} \supseteq \mathcal{P}(C_{\text{prop}} \wedge \neg c)$

Given a node v in t with label $(C_p, C_n, C_{\text{prop}})$, we set $\|v\| = \|C_{\text{prop}}\|$. For every tree t, we denote with $\prec_t$ the partial ordering of nodes induced by t.

Definition 2 (Expanded, C_{Pr}-Complete w.r.t $\mathcal{S}$ and $\mathcal{S}$-Reduced Trees). *The search tree t is* completely expanded *if every leaf $v = (C_p, C_n, C_{\text{prop}})$ satisfies either 1.)* $\|v\| = \{\alpha\}$ and C_{prop} determines $\mathcal{X}$ to α, *or 2.)* $\perp \in \mathcal{P}(C_{\text{prop}})$. *Let $\mathcal{S}$ be a symmetry set for C_{Pr}. A search tree is C_{Pr}-*complete w.r.t. $\mathcal{S}$ *if for every $\alpha \in \|C_{\text{Pr}}\|$ there is a leaf v such that*

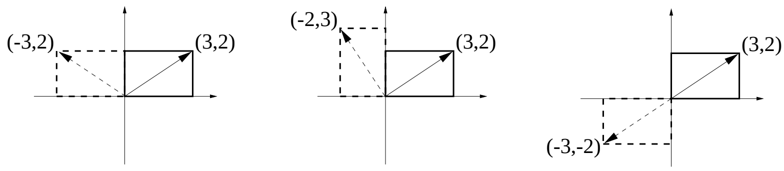
$$\|v\| = \{\alpha\} \;\vee\; \exists s \in \mathcal{S}\backslash\{\text{id}_{C_{\text{Pr}}}\} : \|v\| = \{s(\alpha)\}.$$

*A search tree is $\mathcal{S}$-*reduced *if for every leaf v with $\|v\| = \{\alpha\}$ we have that $\forall s \in \mathcal{S}\ \forall v' \neq v : (\|v'\| = \{\alpha'\} \Rightarrow s(\alpha') \neq \alpha)$.*

In our case, the symmetries are rotations and reflections. These are affine mappings $S : \mathbb{Z}^3 \to \mathbb{Z}^3$ with $S(\boldsymbol{x}) = A_S\boldsymbol{x} + \boldsymbol{v}_S$ that map the $\mathbb{Z}^3$ onto $\mathbb{Z}^3$. I.e., the matrix A_S is an orthogonal matrix with the property that the columns $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ of A_S satisfy $\forall i \in [1..3] : \boldsymbol{v}_i \in \{\pm\boldsymbol{e}_x, \pm\boldsymbol{e}_y, \pm\boldsymbol{e}_z\}$. Since the dimension of A_S must be 3, we have $6 \times 4 \times 2$ matrices, and henceforth 47 non-trivial symmetries. The problem is that the vector $\boldsymbol{v}_S$ is not yet fixed. Now in our case, the use of the frame surrounding the core monomers allows one to fix this vector. As an example, we use $\mathbb{Z}^2$ with a rectangular frame. For every symmetry s, we have to fix $\boldsymbol{v}_S$ such that the frame is mapped to itself. If this is not possible, then the corresponding symmetry is excluded by the frame dimension. Consider a frame in $\mathbb{Z}^2$ with starting point $(0, 0)$ and dimensions $\texttt{Frx} = 4$ and $\texttt{Fry} = 3$.[4] Then the top left point of the frame is $(3, 2)$. Furthermore, consider the three symmetries reflection at the y-axis, rotation by 90° and rotation by 180°, which we will name S_1, S_2 and S_3 in the following. The corresponding matrices are

$$A_{S_1} = \begin{pmatrix} -1\,0 \\ 0\,1 \end{pmatrix} \quad A_{S_2} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad A_{S_3} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2}$$

and the corresponding mappings are



.

A symmetry S is compatible with the frame dimensions $(\texttt{Frx}, \texttt{Fry})$ if the frame is mapped to itself, i.e., if $\{\boldsymbol{v} \mid \boldsymbol{0} \leq \boldsymbol{v} \leq (\texttt{Frx} - 1, \texttt{Fry} - 1)\} = \{S(\boldsymbol{v}) \mid \boldsymbol{0} \leq$

[4] If we define an appropriate symmetry S for a frame with starting point $(0, 0)$, then we get a symmetry for a frame with the same dimension and starting point $\boldsymbol{s}$ by using the affine mapping $S'(\boldsymbol{x}) = S(\boldsymbol{x} - \boldsymbol{s}) + \boldsymbol{s} = S(\boldsymbol{x}) + \boldsymbol{s} - A_S\boldsymbol{s}$.

$\boldsymbol{v} \leq (\mathtt{Frx}-1, \mathtt{Fry}-1)\}$. For a given matrix A_S, there exists a $\boldsymbol{v}_S$ such that $S(\boldsymbol{x}) = A_S\boldsymbol{x} + \boldsymbol{v}_S$ satisfies this condition if and only if A_S satisfies

$$A_S(\mathtt{Frx}-1, \mathtt{Fry}-1) = (a_x, a_y) \quad \text{and} \quad |a_x| = \mathtt{Frx}-1 \wedge |a_y| = \mathtt{Fry}-1. \qquad (3)$$

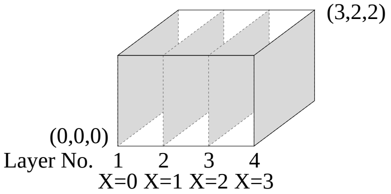
For the matrices A_{S_1}, A_{S_2} and A_{S_3}, we get $(-3,2)$, $(-2,3)$ and $(-3,-2)$, which excludes the symmetry characterised by A_{S_2}.

Given a symmetry characterised by an orthogonal matrix A_S which is compatible according to (3), then $\boldsymbol{v}_S = (v_x, v_y)$ is defined by

$$v_x = \begin{cases} -a_x & \text{if } a_x < 0 \\ 0 & \text{else} \end{cases} \quad \text{and} \quad v_y = \begin{cases} -a_y & \text{if } a_y < 0 \\ 0 & \text{else} \end{cases},$$

where a_x and a_y are defined by (3). The extension to three dimension is straightforward.

Now the symmetries are excluded by adding at the right branch (which is visited after the left branch) constraints which enforce the right branch to exclude all solutions for which a symmetric solution has been found in the left branch. For this purpose, we need the notion of *symmetric constraints*. As an example, we use reflection along the x-axis S^{rx} in three dimensions. Furthermore, assume that we have selected a frame with the dimensions $(\mathtt{Frx}, \mathtt{Fry}, \mathtt{Frz}) = (4,3,3)$ with starting point $(0,0,0)$. Then the frame is of the form



Using the above outlined method, S^{rx} is defined by

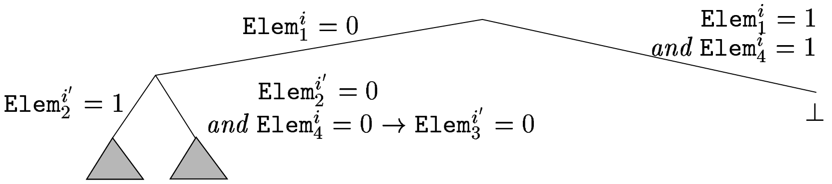
$$S^{\mathrm{rx}}(\boldsymbol{x}) = \begin{pmatrix} -1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} -(-(\mathtt{Frx}-1)) \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix}$$

Now consider the constraint $\mathtt{Elem}_j^i =: b$, where $b \in \{0,1\}$. $\mathtt{Elem}_j^i =: b$ is defined by a reified constraint $\mathtt{Elem}_j^i =: (\mathtt{X}_i =: j-1)$. We first want to calculate the S^{rx}-symmetric constraint $S^{\mathrm{rx}}_{con}(\mathtt{X}_i =: j-1)$. Given some conformation c satisfying the constraint $\mathtt{X}_i =: j-1$, we know that the coordinates of the i^{th} monomer are $(j-1, y_i, z_i)$ for some y_i, z_i. Furthermore, we know that these coordinates are mapped to $S^{\mathrm{rx}}(j-1, y_i, z_i)$ in the S^{rx}-symmetric conformation c' of c. Hence, we know that c' satisfies the constraint $\mathtt{X}_i =: a$, where a is the x-coordinate of $S^{\mathrm{rx}}(j-1, y_i, z_i)$. Since the x-coordinate of $S^{\mathrm{rx}}(x,y,z)$ is $-x+3$, we can conclude that the symetric constraint $S^{\mathrm{rx}}_{con}(\mathtt{X}_i =: j-1)$ is $\mathtt{X}_i =: 3-(j-1)$, which is equivalent to $\mathtt{X}_i =: 4-j$. Now we can use this to define the symmetric constraint $S^{\mathrm{rx}}_{con}(\mathtt{Elem}_j^i =: b)$ for $\mathtt{Elem}_j^i =: b$. Since $\mathtt{X}_i =: 4-j$ is equivalent to

$\mathtt{X}_i =: (5-j)-1$, and $\mathtt{X}_i =: k-1$ is equivalent to $\mathtt{Elem}_k^i =: 1$, we get that the S^{rx}-symmetric constraint to $\mathtt{Elem}_j^i =: b$ is

$$\mathtt{Elem}_{5-j}^i =: b.$$

This states if the i^{th} H-monomer is in the 1^{st} layer of the frame, then the i^{th} H-monomer must be in the 4^{th} layer in the conformation produced by S^{rx}. Using this construction for generating symmetric constraints, we can present an example of a (partial) $\{S^{\mathrm{rx}}\}$-excluded search tree. Here, the constraints added by the symmetry exclusion algorithm are indicated by a leading *and*:



$\mathtt{Elem}_1^i = 0$
$\mathtt{Elem}_1^i = 1$
and $\mathtt{Elem}_4^i = 1$
$\mathtt{Elem}_2^{i'} = 1$
$\mathtt{Elem}_2^{i'} = 0$
and $\mathtt{Elem}_4^i = 0 \rightarrow \mathtt{Elem}_3^{i'} = 0$
$\perp$

In the right-most branch, we have added the constraint $\mathtt{Elem}_4^i = 1$, which is the same as $\neg S_{con}^{\mathrm{rx}}(\mathtt{Elem}_1^i = 0)$. Together with $\mathtt{Elem}_1^i = 1$, this yields an immediate contradiction. The reason is simply the following. Consider any conformation satisfying $\mathtt{Elem}_1^i = 1$ (the label of the right-most branch). Then we know that the monomer i is in the 1^{st} layer. Consider an arbitrary conformation c which is generated from a conformation c' satisfying $\mathtt{Elem}_1^i = 1$ by reflection at the x-axis. Then c has monomer i in the 4^{th} layer, and henceforth satisfies $\mathtt{Elem}_4^i = 1$. But $\mathtt{Elem}_4^i = 1$ implies $\mathtt{Elem}_1^i = 0$, which implies that c was already found in the left branch. Henceforth, the symmetry exclusion closes the right-most branch.

2.4 A new lower bound

We will now describe a lower bound on the surface provided that know the distribution of H-monomers to x-layers. For the rest of this section, let $\mathtt{E}_j\mathtt{.seh}$ (resp. $\mathtt{E}_j\mathtt{.soh}$) be the number of even (resp. odd) H-monomers in the j^{th} x-layer. Given a conformation c, we distinguish between x-surface and yz-surface of c. The x-surface of c is defined by

$$\mathrm{Surf}_s^x(c) = \left|\left\{\, (c(i), \boldsymbol{p}) \,\middle|\, s_i = H \ \wedge\ \boldsymbol{p} - c(i) = \pm \boldsymbol{e}_x \wedge \forall j : (s_j = H \Rightarrow c(j) \neq \boldsymbol{p}) \,\right\}\right|$$

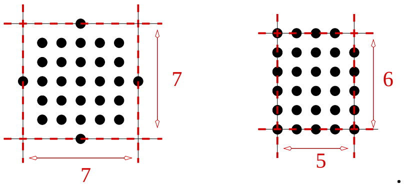
The yz-surface of c is just $\mathrm{Surf}_s(c) - \mathrm{Surf}_s^x(c)$. For the lower bounds on x-surface and yz-surface, we use a special property of the cubic lattice, namely that even H-monomers can form contacts only with odd H-monomers. Given a point $(x,y,z) \in \mathbb{Z}^3$, we say that (x,y,z) is *odd* (resp. *even*) if $x+y+z$ is odd (resp. even). We write $(x,y,z) \equiv (x',y',z')$ iff $x+y+z \equiv x'+y'+z' \mod 2$. Then we have for every conformation c of s that $c(i) \equiv c(j)$ iff $i \equiv j \mod 2$. Using this property, we get the following a lower bound on the x-surface:

$$\begin{aligned}\mathrm{Surf}_s^x(c) \geq\ & \mathtt{E}_1\mathtt{.soh} + \mathtt{E}_1\mathtt{.seh} + \mathtt{E}_{\mathtt{Frx}}\mathtt{.soh} + \mathtt{E}_{\mathtt{Frx}}\mathtt{.seh} \\ & + \sum_{1 \leq j < \mathtt{Frx}} \left(|\mathtt{E}_j\mathtt{.soh} - \mathtt{E}_{j+1}\mathtt{.seh}| + |\mathtt{E}_j\mathtt{.seh} - \mathtt{E}_{j+1}\mathtt{.soh}|\right)\end{aligned}$$

For a lower bound on the yz-surface, we consider the surface contribution in the different x-layers. Now let the j^{th} x-layer be defined by the equation $x = a_j$, and let $P(x = a_j)$ be the set of points in the plane $x = a_j$. We define the yz-surface of this layer by

$$\mathrm{Surf}^{\mathrm{s}}_{\mathrm{j}}(\mathrm{c}) = \left| \left\{ (c(i), \boldsymbol{p}) \in P(x = a_j)^2 \;\middle|\; \begin{array}{l} s_i = H \ \wedge\ \boldsymbol{p} - c(i) \in \{\pm \boldsymbol{e}_y, \pm \boldsymbol{e}_z\} \\ \wedge\ \forall j : (s_j = H \Rightarrow c(j) \neq \boldsymbol{p}) \end{array} \right\} \right|$$

The first lower bound is given in [18], where it was found that the surface in layer j is given by the minimal rectangle enclosing the H-monomers in that layer. Thus, consider the following two conformations, where the positions occupied by H-monomers in the j^{th} x-layer look as follows:



.

Both have the property that $\mathtt{E}_j\mathtt{.soh}+\mathtt{E}_j\mathtt{.seh} = 29$. But $\mathrm{Surf}^{\mathrm{s}}_{\mathrm{j}}(\mathrm{c})$ is $2\cdot 7+2\cdot 7 = 28$ for the first conformation, and $2 \cdot 5 + 2 \cdot 6 = 22$ for the second. Hence, given $n_H = \mathtt{E}_j\mathtt{.soh} + \mathtt{E}_j\mathtt{.seh}$, then a lower bound for $\mathrm{Surf}^{\mathrm{s}}_{\mathrm{j}}(\mathrm{c})$ is given by $2 \cdot a + 2 \cdot b$, where $a = \lceil \sqrt{n_H} \rceil$ and $b = \lceil \frac{n_H}{a} \rceil$.

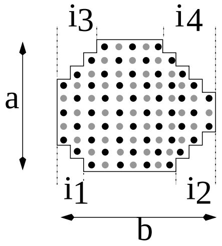
But we can provide a better lower bound for $\mathrm{Surf}^{\mathrm{s}}_{\mathrm{j}}(\mathrm{c})$ by considering the different parity of H-monomers. For this purpose, we introduce the concept of a coloring as an abstraction of the points occupied by H-monomers in a conformation c. A *coloring* is a function $f : \mathbb{Z}^2 \to \{0, 1\}$. We say that a point (x, y) is colored black by f iff $f(x, y) = 1$. In the following, we consider only colorings different from the empty coloring f_e (which satisfies $\forall \boldsymbol{p} : f_e(\boldsymbol{p}) = 0$). A point $(x, y) \in \mathbb{Z}^2$ is a *caveat in* f if (x, y) is contained in the hull (over $\mathbb{Z}^2$) of the points colored black in f. Given a coloring f, define $e(f) = |\{(x, y) \mid f(x, y) = 1 \text{ and } x + y \text{ even}\}|$ and $o(f) = |\{(x, y) \mid f(x, y) = 1 \text{ and } x + y \text{ odd}\}|$. The *surface* $\mathrm{Surf}(f)$ of a coloring f is defined analogously to the surface of a conformation, i.e., it is the number of pairs where the first point is colored black by f, and the second is colored white. Given a pair (e, o) of integers, we define $\mathrm{Surf}(e, o)$ to be $\min\{\mathrm{Surf}(f) \mid f \text{ colouring with } e(f) = e \wedge o(f) = o\}$. W.l.o.g, we can restrict ourself to cases where $e \leq o$. Thus, we have the following lemma.

Lemma 1. $\mathrm{Surf}^{\mathrm{s}}_{\mathrm{j}}(\mathrm{c}) \geq \begin{cases} \mathrm{Surf}(\mathtt{E}_j\mathtt{.seh}, \mathtt{E}_j\mathtt{.soh}) \ \textit{if}\ \mathtt{E}_j\mathtt{.seh} \leq \mathtt{E}_j\mathtt{.soh} \\ \mathrm{Surf}(\mathtt{E}_j\mathtt{.soh}, \mathtt{E}_j\mathtt{.seh}) \ \textit{if}\ \mathtt{E}_j\mathtt{.soh} \leq \mathtt{E}_j\mathtt{.seh}. \end{cases}$

In the following theorem, we handle the simple case where $|e - o| \leq 1$. There, the lower bound on colorings agrees with the lower bound as given in [18].

Theorem 1. *Let (e, o) be a pair of integers with $|e - o| \leq 1$. Let $a = \lceil \sqrt{e + o}\, \rceil$ and $b = \lceil \frac{e+o}{a} \rceil$. Then* $\mathrm{Surf}(e, o) = 2a + 2b$.

The remaining case is to calculate $\mathrm{Surf}(e, o)$ where $e < o + 1$. But it would be too time consuming to search through all possible colorings f in order to determine $\mathrm{Surf}(e, o)$. But this is not necessary, since we can consider a 'normal form' of colorings to which every coloring can be extended. The normal forms are kind of maximal colorings provided a given difference $d(f) = o(f) - e(f)$. We will handle only caveat-free colorings for simplicity reasons. Let f be a coloring. Then we define $\mathrm{length}(f)$ to be $\max\{|x - x'| \mid \exists y, y' : f(x, y) = 1 = f(x', y')\} + 1$, and $\mathrm{height}(f)$ to be $\max\{|y - y'| \mid \exists x, x' : f(x, y) = 1 = f(x', y')\} + 1$. The pair $(\mathrm{height}(f), \mathrm{length}(f))$ is called the *frame* of f. We define the partial order $\preceq$ on caveat-free colorings by $f \preceq f'$ if and only if $\mathrm{height}(f) = \mathrm{height}(f')$, $\mathrm{length}(f) = \mathrm{length}(f')$ and $d(f) = d(f')$. It is easy to see that $\mathrm{Surf}(f) = \mathrm{Surf}(f')$ given $f \preceq f'$. We can show that every f can be extended to a $\preceq$-maximal coloring f' (which must have the same surface). Furthermore, we can show, that every $\preceq$-maximal coloring f has a simple form. An example of a $\preceq$-maximal coloring f with $o(f) > e(f)$ is



Here, we use black beads for odd positions (x, y) with $f(x, y) = 1$, and grey beads for even positions (x, y) with $f(x, y) = 1$. (a,b) is the frame of f, and $i_1, \ldots, i_4$ are the side length of triangles excluded at the corner. The tuple $(a, b, i_1, i_2, i_3, i_4)$ is called the characteristics of this coloring. In this case, the characteristics is $(10, 12, 2, 3, 3, 4)$.

Theorem 2. *Let f be a $\preceq$-maximal coloring. Then f has a unique characteristics $(a, b, i_1, i_2, i_3, i_4)$. Furthermore, we have $e(f) + o(f) = a \times b - \sum_{j=1}^{4} \frac{i_j(i_j+1)}{2}$, $d(f) = \frac{i_1+i_2+i_3+i_4}{2} + 1$ and $\mathrm{Surf}(f) = 2a + 2b$.*

3 Results

We have tested the program on all sequences presented in [18]. For all we found an optimal conformation. In Table 1, we have listed the test sequences together with the found optimal conformation, the sequence length and the optimal surface. For comparison, the runtimes (on a Sun4) of the algorithm in [18] for all optimal conformations are 1 h 38 min for L1, 1 h 14 min for L2, 5 h 19 min for L3, 5 h 19 min for L4 and 20 min for L5, respectively. There is a newer, more efficient version of this algorithm reported in [19], but there are no explicit runtime given for these or others sequences. In Table 2, we have listed the number of steps to find a first conformation (and a second, if the first was not optimal), the number of steps needed to prove optimality, and the runtime on a Pentium 180 Pro.

Sequence and Sample Conformation	Length	Optimal Surface
L1 HPPPPHHHHPPHPHPHHHPHPPHHPPH	27	40
RFDBLLFRFUBULBDFLUBLDRDDFU		
L2 HPPPHHHHPHPHHPPPHPHHPHPPPHP	27	38
RFDLLBUURFDLLBBRURDDFDBLUB		
L3 HPHHPPHHPPHHHHPPPHPPPHHHPPH	27	38
RFLDLUBBUFFFDFURBUBBDFRFDL		
L4 HHPHHPHHPHHHHHHPPHHHHHHPPHHHHHHH	31	52
RRFDBLDRFLLBUFLURFDDRFUBBUFRDD		
L5 PHPPHPPHPPHPPHPPHPPHPPHPPHPPHPPHPPHP	36	32
RFDBDRUFUBRBLULDLDRDRURBLDLULURBRFR		

Table 1. Test sequences. Below every sequence, we list an optimal conformation represented as a sequence of bond directions (R=right,L=left and so on).

Acknowledgement

I would like to thank Prof. Peter Clote, who got me interested in bioinformatics, and enabled and inspired this research. I would like to thank Prof. Martin Karplus for helpful discussions on the topic of lattice models, and for motivating me to apply constraint programming techniques to lattice protein folding. I would like to thank Dr. Erich Bornberg-Bauer, who initiated this research, too. I would like to thank him also for explaining me the biological background, and for many discussion and hints. Furthermore, I would like to thank Sebastian Will, who contributed on the section 'Exclusion of Geometrical Symmetries'.

References

1. V. I. Abkevich, A. M. Gutin, and E. I. Shakhnovich. Impact of local and non-local 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. 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.
4. 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.
5. 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.
6. 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.
7. Ken A. Dill, Klaus M. Fiebig, and Hue Sun Chan. Cooperativity in protein-folding kinetics. *Proc. Natl. Acad. Sci. USA*, 90:1942 – 1946, 1993.
8. 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.

Seq.	1^{st} Conf.		2^{nd} Conf.		total # Steps	Runtime
	# Steps	Surface	# Steps	Surface		
L1	519	40 (opt.)	—	—	921	3.85 sec
L2	1322	40	1345	38 (opt.)	5372	1 min 35 sec
L3	1396	38 (opt.)	—	—	1404	4.09 sec
L4	35	52 (opt.)	—	—	38	0.68 sec
L5	1081	32 (opt.)	—	—	1081	4.32 sec

Seq.	1^{st} Conf.		2^{nd} Conf.		total # Steps	Runtime
	# Steps	Surface	# Steps	Surface		
L1	139	40 (opt.)	—	—	159	3.35 sec
L2	43	38 (opt.)	—	—	61	1.53 sec
L3	217	38 (opt.)	—	—	218	1.17 sec
L4	28	52 (opt.)	—	—	28	1.05 sec
L5	25	32 (opt.)	—	—	25	440 ms

Table 2. Search time and number of search steps for the sample sequences. The first table contains the results for an implementation without symmetry exclusion, the second table for the current implementation containing symmetry exclusion. The main reduction in the number of search steps is due to the symmetry exclusion. Only for L4, the older implementation achieves a better result. The reason is that for this sequence, both implementations actually do not have to perform a search to find the optimal conformation. In this case, the symmetry exclusion is clearly an overhead.

9. S. Govindarajan and R. A. Goldstein. The foldability landscape of model proteins. *Biopolymers*, 42(4):427–438, 1997.
10. William E. Hart and Sorin C. Istrail. Fast protein folding in the hydrophobid-hydrophilic model within three-eighths of optimal. *Journal of Computational Biology*, 3(1):53 – 96, 1996.
11. David A. Hinds and Michael Levitt. From structure to sequence and back again. *Journal of Molecular Biology*, 258:201–209, 1996.
12. 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.
13. 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.
14. 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.
15. R. Unger and J. Moult. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, 231:75–81, 1993.
16. Ron Unger and John Moult. Local interactions dominate folding in a simple protein model. *Journal of Molecular Biology*, 259:988–994, 1996.
17. A. Šali, E. Shakhnovich, and M. Karplus. Kinetics of protein folding. *Journal of Molecular Biology*, 235:1614–1636, 1994.
18. Kaizhi Yue and Ken A. Dill. Sequence-structure relationships in proteins and copolymers. *Physical Review E*, 48(3):2267–2278, September 1993.
19. Kaizhi Yue and Ken A. Dill. Forces of tertiary structural organization in globular proteins. *Proc. Natl. Acad. Sci. USA*, 92:146 – 150, 1995.