A Constraint-Based Approach to Structure Prediction for Simplified Protein Models that Outperforms Other Existing Methods

Rolf Backofen and Sebastian Will

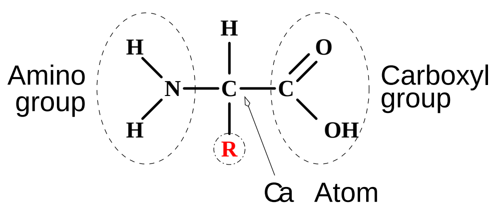
Friedrich-Schiller-Universitaet Jena
Institute of Computer Science, Chair for Bioinformatics
Ernst-Abbe-Platz 1-4, D-07743 Jena, Germany,
backofen@inf.uni-jena.de

Abstract Lattice protein models are used for hierarchical approaches to protein structure prediction, as well as for investigating general principles of protein folding. So far, one has the problem that either the lattice does not model real protein conformations with good quality, or there is no efficient method known for finding native conformations.
We present a constraint-based method that largely improves this situation. It outperforms all existing approaches in lattice protein folding on the type of model we have chosen (namely the HP-model by Lau and Dill[34], which models the important aspect of hydrophobicity). It is the only exact method that has been applied to two different lattices. Furthermore, it is the only exact method for the face-centered cubic lattice. This lattice is important since it has been shown [38] that the FCC lattice can model real protein conformations with coordinate root mean square deviation below 2 Å.
Our method uses a constraint-based approach. It works by first calculating maximally compact sets of points (hydrophobic cores), and then threading the given HP-sequence to the hydrophobic cores such that the core is occupied by H-monomers.

1 Introduction

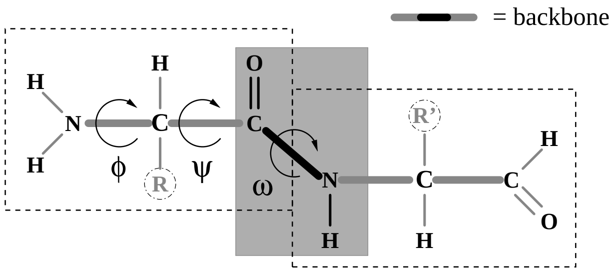
1.1 Proteins

Proteins are sequences composed of an alphabet of 20 amino acids. An amino acid is a chemical group of the form



where R is a chemical group (called *chain residue*) specifying the type of the amino-acid. The central carbon atom is the α-*carbon* (short $C\alpha$), the left NH_2 groups is the *amino group*, and the left COOH the *carboxy group*. There are 20 different chain residues, which have different chemical properties. Residues can be hydrophobic or hydrophilic, small or large, charged or uncharged.

Two amino acids can be connected via a *peptide bond*, where the carboxy group of the first amino acid reacts with the amino group of the second. The result is a group of the form



(1)

Using the peptide bond, long sequences of amino acids (i.e., proteins) can be generated.

The peptide bond itself (indicated with a grey rectangle in (1)) is usually planar, which means that there is no free rotation around this bond.[1] There is more flexibility for rotation around the N-$C\alpha$-bond (called the ϕ-angle), and around the $C\alpha$–C bond (called the ψ-angle). But even there, the allowed values of combinations of ϕ and ψ angles are restricted to small regions in natural proteins (which are displayed on so-called Ramachandran plots).

Using this freedom of rotation, the protein can *fold* into a specific three-dimensional structure (also called *conformation*). For natural proteins, the final structure that is achieved is uniquely determined by the sequence of amino acids. For this reason, one speaks of the *native structure* of a given amino acid sequence. The native structure itself uniquely determines the function of the protein.

1.2 Simplified Models of Proteins

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-completeness of the problem has been proven for many different models (including lattice and off-lattice models) [17,21]. 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

[1] There are two conformations for the peptide bond, namely *trans* (corresponding to a rotation angle of 180°), and *cis* (corresponding to a rotation angle of 0°). The *cis* conformation is rare and occurs usually in combination with a specific amino acid, namely Proline.

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, where one faces the difficulty to design proteins that have a unique and stable native structure.

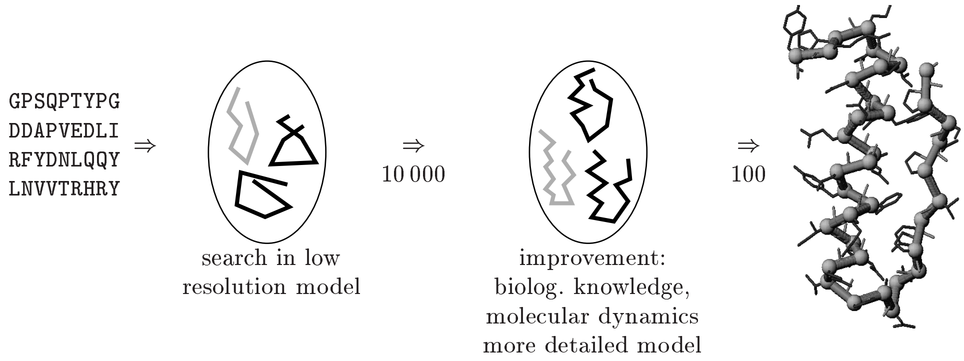


Figure 1. Hierarchical Approach to Protein Structure Prediction

To tackle structure prediction and related problems simplified models have been introduced. These simplified models have been successfully used by several groups in the international contest on automated structure prediction (see the meeting review of CASP3 [33]). They are used in hierarchical approaches for protein folding [46] as follows (see also Figure 1). Given a protein sequence, then one first enumerates all low (or minimal) energy conformations in a simplified model. In the simplified model, only some aspects of the protein structure are modeled (for which reason they are also called low-resolution or coarse-grained protein models). Then, the say 10 000 best structures are taken and fine-tuned using other methods. These methods usually incorporate biological knowledge and simulation of protein folding on full atomic detail (i.e. molecular dynamics simulation).

The most important class of simplified models 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. Apart from their use in structure prediction, they have became a major tool for investigating general properties of protein folding. They constitute a genotype (protein sequence) versus phenotype (protein conformation) mapping that can be dealt using computational methods. Thus, they can be used to investigate evolutionary processes. An example is [18], where so-called neutral networks have been investigated. The edges of the network are pairs of sequences which differ only in one positions, but have the same minimal energy conformation. Thus, a neutral network represents all protein sequences encoding the same protein conformation. The question is whether one can switch between two different neutral networks using only a small number of amino-acid substitutions. If this is the case, then this suggest a way evolution could have produced the diversity of protein conformations found in nature.

In the literature, many different lattice models, (i.e., lattices and energy functions) have been used (see related work). Of course, the question arises which lattice and energy functions has to be preferred. There are two (somewhat conflicting) aspects that have to be evaluated when choosing a model: 1) the accuracy of the lattice in approximating real protein conformations, and the ability of the energy function to discriminate native from non-native conformations, and 2) the availability and quality of search algorithm for finding minimal (or low) energy conformations.
While the first aspect is well-investigated in the literature (e.g., [38,23]), the second aspect is underrepresented. By and large, there are mainly two different heuristic search approaches used in the literature: 1) Ad hoc restriction of the search space to compact or quasi-compact conformations. A good example is [43], where the search space is restricted to conformations forming an $n \times n \times n$-cube. The main drawback here is that the restriction to compact conformation is not biologically motivated for a complete amino acid sequence (as done in these approaches), but only for the hydrophobic amino acids. In consequence, the restriction either has to be relaxed and then leads to an inefficient algorithm, or is chosen to strong and thus may exclude minimal conformations. 2.) Stochastic sampling like Monte Carlo methods with simulated annealing, genetic algorithms etc. Here, the degree of optimality for the best conformations and the quality of the sampling cannot be determined by state of the art methods.[2]

In this work, we introduce a constraint-based approach that completely outperforms existing approaches in lattice protein folding (on the type of model we have chosen). It outperforms them even in several ways, namely flexibility, completeness and efficiency.

1.3 Contribution of the Paper: A Constraint-based approach

We have considered the HP-model, which is an important representative of lattice models. It has been introduced by Lau and Dill in [34]. 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 2(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* (also called *optimal conformation*) is just a conformation with the maximal number of contacts between H-monomers. Just recently, the

[2] Despite there are mathematical treatments of Monte Carlo methods with simulated annealing, the partition function of the ensemble (which is needed for a precise statement) is in general unknown.

(a)

	H	P
H	-1	0
P	0	0

(b)

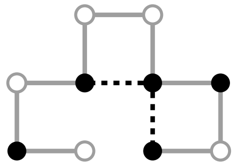


Figure 2. Energy matrix and sample conformation for the HP-model

structure prediction problem has been shown to be NP-complete even for the HP-model [17,21].

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

Concerning flexibility, our method is the only one that works for two different but important lattices. These are the cubic lattice and the face-centered-cubic lattice (FCC). The cubic lattice is important for comparability to other existing approaches since it is the lattice which is explored most. However, the ability of this lattice to approximate real protein conformations is poor. For example, [4] pointed out especially the parity problem in the cubic lattice. This drawback of the cubic lattice is that every two monomers with chain positions of the same parity cannot form a contact.

The FCC on the other hand is important since it can model real protein conformations with good quality (see [38], where it was shown that FCC can model protein conformations with coordinate root mean square deviation below 2 Å). Recently, [15,16] have shown that neighborship of amino acids in proteins closely resembles a distorted FCC-lattice, and that the FCC is best suited for modeling proteins. This is an immediate effect of hydrophobic packing. Just recently, it was shown that the FCC is the lattice allowing the densest packing of identical spheres, approximately 400 years after the original conjecture by Kepler [40,20] (see Figure 3 for a description of the FCC lattice).

Concerning completeness, we have applied the method to existing structures and found additional optimal conformations not found by other existing approaches. Only for the HP-model on the cubic lattice, there is one other method that can provably find and enumerate optimal conformations [49]. In the HP-model, a sequence usually does not have a unique minimal energy conformation. Instead, there are many minimal energy conformation, and [49] have given a lower bound for the number of such conformations for specific sequences. We have applied our approach to this problem and strongly improved this lower bound.

Concerning efficiency, we have successfully applied our algorithm to sequences up to length 200 in the face-centered cubic lattice (FCC). For these sequences, we can provably find a minimal energy conformation. For the FCC, so far there existed only heuristic algorithms (for an example of a genetic algorithm for arbitrary Bravais lattices see [14]). Usually, these algorithms are applied to sequence of say length at most 80 (where they usually find only a low but *not* minimal energy conformation). Since the search space for conformations in the cubic lat-

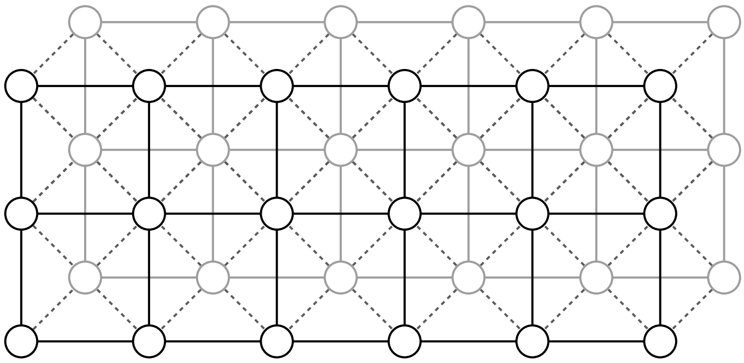


Figure 3. Two layers of the face-centered-cubic lattice (FCC). It can be seen as two layers of the cubic lattice, which are arranged in such a way that every position of the second layer has contacts to 4 positions in the previous layer and vice versa (shown as dashed lines). Hence, every position has 12 neighbors in the FCC.

tice grows with approximately 4.5^n (where n is the length of the sequence), this implies that our method handles a search space that at least by the factor 4.5^{120} higher than the search space handled by other methods for the face-centered cubic lattice.[3]

1.4 Related Work

A discussion of lattice proteins can be found in [23]. 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 [44,1,25,42,31,28,2,37]. Most of them use heuristic methods, ranging from Monte-Carlo simulated annealing (e.g. [35,25]) to genetic algorithms (e.g. [42]), purely heuristic methods like hydrophobic zipper [24] and the chain growth algorithm [19], as well as complete enumeration (often restricted to subset of all conformations, e.g. [43,46]).

First steps have been made to improve the situation on the algorithmic part. The first improvement was the introduction of an exact algorithm for find minimal energy conformations in the cubic HP-Model [49]. The algorithm is called CHCC for "**C**onstraint **H**ydrophobic **C**ore **C**onstruction) (albeit it doesn't use constraint-based methods). This approach works only for the cubic lattice, but not for the FCC-lattice which is much better suited for modeling real protein conformations. The second improvement is the appearance of a bunch of approximation algorithms [30,3] for different lattice models. Also they are a very

[3] The number 4.5^n has been estimated for cubic lattice [36]; for the FCC, we do not know of any good estimation of the number of conformations. But one knows that it must be higher than the number of conformations in the cubic lattice.

important first step in the right direction, their approximation ratio is still not good enough to be used in practice.

Another interesting approach is to combine secondary structure predictions with lattice models. The secondary structure consists of local structure motifs like α-helices and β-sheets. One searches for conformations in the lattice model with low energy having the predicted secondary structure elements. Here, major improvements have been achieved using again a constraint-based approach and the FCC lattice [26].

2 Overview over the Algorithm

2.1 Formal Model

The original HP-model is defined for the cubic lattice. But it is easy to use the HP-energy function for other lattices. In [4], this has been done for the face-centered cubic lattice (FCC). For this reason, we have applied our approach to the HP-model in both the cubic and face-centered cubic lattice. For simplicity, we introduce the formal model for the cubic lattice only. The description for the face-centered-cubic is analogous if one applies the appropriate transformation (for details see [7]). 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

$$c : [1..|s|] \to \mathbb{Z}^d$$

(where $d = 2$ or $d = 3$ depending on whether we consider a 2-dimensional or a 3-dimensional lattice) such that

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

The first condition is imposed by the lattice constraint and implies that the distance vector between two successive elements must be a unit-vector (or a negative unit-vector) in every admissible conformation. The second condition is the constraint that the conformation must be self-avoiding.

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

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

(in other words, the number of pairs of H-monomers that have distance 1 in the conformation c, but are not successive in the sequence s). The energy of c is just $-\mathrm{Contact}_s(c)$. With $\boldsymbol{e}_x$, $\boldsymbol{e}_y$ and $\boldsymbol{e}_z$ we denote the unit vectors $(1, 0, 0)$, $(0, 1, 0)$ or $(0, 0, 1)$, respectively. We say that two points $\boldsymbol{p}, \boldsymbol{p}' \in \mathbb{Z}^3$ are *neighbors* if $\|\boldsymbol{p} - \boldsymbol{p}'\| = 1$. This is equivalent to the proposition that $\boldsymbol{p} = \boldsymbol{p}' \pm \boldsymbol{e}$ with $\boldsymbol{e} \in \{\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z\}$.

This can now be directly encoded as a constraint problem. Our constraint problem consists of finite domain variables. We use also Boolean constraint and reified constraints. With *reified constraints* we mean a constraint $\mathtt{x} =: (\phi)$, where ϕ is a finite domain constraint. $\mathtt{x}$ is a Boolean variable which is 1 if and only if ϕ holds. Technically, this can be achieved via setting x to 1 if the constraint store entails ϕ, and to 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 $\phi \to \psi$, 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 [41], which supports finite domain variables, Boolean constraints, reified constraints, entailment constraints and a programmable search module.

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 these coordinates are all integers. But we can even restrict the possible values of these variables to the finite domain $[1..2n]$.[4] This is expressed by introducing the constraints

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

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 $i \neq j$.[5]

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

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

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.

The most simplest way to search for conformations with maximal number of contacts would be to add constraints for counting the number of contacts. Then one can directly enumerate the variables $\mathtt{X}_i$, $\mathtt{Y}_i$ and $\mathtt{Z}_i$. For HP-type models, we have to count contacts which are always generated between two neighboring H-monomers. For this purpose, one introduces a variable $\mathtt{Contact}_{i,j}$ that is 1

[4] We even could have used $[1..n]$. But the domain $[1..2n]$ 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.

[5] This cannot be directly encoded in Oz [41], but we reduce these constraints to difference constraints on integers.

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

$$\begin{aligned} \mathtt{Xdiff}_{i,j} &= |\mathtt{X}_i - \mathtt{X}_j| \qquad \mathtt{Zdiff}_{i,j} = |\mathtt{Z}_i - \mathtt{Z}_j| \\ \mathtt{Ydiff}_{i,j} &= |\mathtt{Y}_i - \mathtt{Y}_j| \qquad \mathtt{Contact}_{i,j} \in \{0,1\} \\ (\mathtt{Contact}_{i,j} = 1) &\leftrightarrow (\mathtt{Xdiff}_i + \mathtt{Ydiff}_i + \mathtt{Zdiff}_i = 1) \end{aligned} \tag{2}$$

where $\mathtt{Xdiff}_{i,j}$, $\mathtt{Xdiff}_{i,j}$ and $\mathtt{Zdiff}_{i,j}$ are new variables. The variable $\mathtt{HHContacts}$ counts the number of contacts between H-monomers, and is defined by

$$\mathtt{HHContacts} = \sum_{\substack{i+1<j\wedge \\ s(i)=H\wedge s(j)=H}} \mathtt{Contact}_{i,j}. \tag{3}$$

Now we could start to apply constraint-based enumeration on $\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i$ searching for a conformation with maximal number of contacts.

The main problem using this approach alone is that it is very difficult to define good bounds and to find a search heuristic for enumerating low-energy conformation first. Nevertheless, this formulation is in part required for lattice models with an extended alphabet like the HPNX-model [13], which models also electrostatic contacts in addition to hydrophobicity.

To overcome this problem, one has to find a way to restrict the set of all conformations in a pre-processing step to a subset of conformations that contains provably all minimal energy conformations. For this purpose, we apply a technique that has been used first by the CHCC algorithm [48]. There, one calculates first the so-called hydrophobic cores, which are the positions occupied by H-monomers. Formally, a *hydrophobic core* $\mathcal{C}$ is just a set of positions. The number of contacts in a hydrophobic core $\mathcal{C}$ is defined by

$$\mathtt{Contact}(\mathcal{C}) = \frac{1}{2}|\{(\boldsymbol{p}, \boldsymbol{p}') \mid \boldsymbol{p}, \boldsymbol{p} \in \mathcal{C} \wedge \boldsymbol{p} \text{ and } \boldsymbol{p} \text{ are neighbors}\}|$$

Of course, one has to consider all possible hydrophobic cores with a sufficient number of contacts (where generating the cores is a problem by its own). Luckily, the number of cores that have to be visited is usually feasible and can be precalculated, especially if symmetric solutions are discarded. In [9,12], we have introduced the first general method for breaking symmetries in constraint-based search, which has now become sort of a standard method.

For each core, one has then to find a conformation of the sequence considered that has exactly the corresponding core. This process is called to *thread* the sequence onto the core. We have found good constraint formulations to solve the threading problem (which will be described in section 5). Of course, if we start with a core that is maximally compact (i.e., has the maximal number of contacts between the positions occupied by H-monomers), then it is clear that a conformation generated by threading the sequence onto this core must be a minimal energy conformation. If one starts with all maximally compact cores,

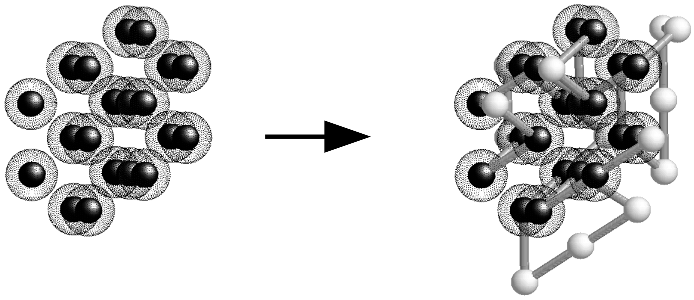


Figure 4. The threading step for the HP-sequence 'HHPPPPHHHHPHHHPHHHP-PHHPHHPPPHHHPH'. The sequence contains 20 H-monomers. To the left a maximally compact hydrophobic core for 20 H-monomers is shown. The core is simply a set of 20 positions occupied by H-monomers. To the right, a conformation for this sequence having the select core is shown. H-monomers are indicated as black beads, and P-monomers as white beads.

and continue with the next best cores, then it is clear that we provably find the minimal energy conformations this way.

The remaining problem is to generate the hydrophobic cores. Without any additional efforts, we would have a huge search space again, since nothing is know about the shape of the hydrophobic core. Hence, one has to find again some restriction on the search space that allows us to calculate bounds and to apply constraint-bases optimization (i.e., branch-and-bound together with constraint propagation). The first idea for restricting the hydrophobic core is to define the surrounding cuboid that contains the hydrophobic core. If one has a very tight cuboid, then it is clear that the hydrophobic core in this cuboid must be very compact (at least in the cubic lattice, in the FCC it is more complicated).

Albeit this helps for the compact conformations, this approach is not fine-grained enough for the FCC as well as for sub-optimal hydrophobic cores in the cubic lattice. We introduce therefore a more precise boundary for the hydrophobic core, which is generated by splitting the lattice into layers. A layer is just a plane orthogonal to the x-dimension. For each layer, we define the *frame* to be the minimal rectangle around all positions of the core in this layer. The corresponding *frame sequence* consists of the height and width of each frame in each layer, together with the number of H-monomers in this layer (see Figure 5). Please note that the exact position of the frames is **not** part of the frame sequence. The sequence of number $n_1 \dots n_l$ will be called *number sequence* in the following.

For finding all hydrophobic cores, one enumerates all frame sequences with the property that $\sum_i n_i$ is the number of H-monomers in the HP-sequence s. In addition, we have to enumerate in principle all possible ways of combining the frames for every frame sequence. Of course, we apply a branch-and-bound

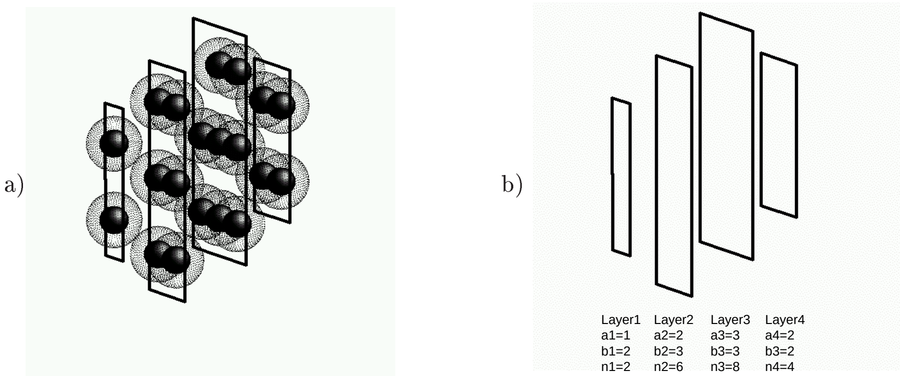
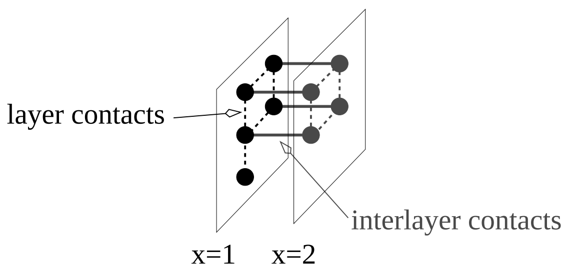


Figure 5. Hydrophobic cores and frame sequences. a) a hydrophobic core with frames. b) the corresponding frame sequence. a_i is the width and b_i is the height of the frame in the i-th layer. And n_i is the number of H-monomers in this layer.

approach to discard combinations which do not yield a sufficient number of contacts. These bound are of course lattice specific and will be described in the next section. In the following section, we will describe the construction of the hydrophobic cores, followed by the description of the threading method.

3 The Lattice-Specific Part: Upper Bound for Frame Sequences

a)



b)

Layer1	Layer2
$a_1 = 2$	$a_2 = 2$
$b_1 = 3$	$b_2 = 2$
$n_1 = 5$	$n_2 = 4$

Figure 6. a) Layer and Interlayer Contacts b) Corresponding Frame Sequence

We will now consider the problem of enumerating the frame sequence using branch-and-bound, starting with the cubic lattice. We need some definitions first. In Figure 6a), a the hydrophobic core for the cubic lattice is shown together with the two layers. All contacts between positions in the same layer are called *layer* *contacts*. All contacts between positions in successive layers are called *interlayer contacts*. To give an upper bound for a specific frame sequence, one give separate bounds for the number of layer *and* interlayer contacts for hydrophobic cores have this frame sequence.

For the layer contacts, consider a frame of size $a \times b$ with n H-monomers. For finding the maximal number of layer contacts that any hydrophobic core with this frame can have, Yue and Dill [48] observed that it is much simpler to calculate the surface instead of the number of layer contacts. The *layer surface* of an hydrophobic core $\mathcal{C}$ in layer $x = k$ is the number of positions $\boldsymbol{p}$ in layer $x = k$ that are not in $\mathcal{C}$, but neighbors of some position $\boldsymbol{p}' \in \mathcal{C}$ ($\boldsymbol{p}$ is called a *surface point*). Since every position in the core has 4 neighbors, which are filled by another member of the core or by a surface point, it is clear that surface and contacts are related via the equation

$$4n = 2\texttt{Contact} + 2a + 2b.$$

Hence, minimizing the surface maximizes the number of contacts.

Now whenever we have a layer where a surface point is buried between two position from the core, this core cannot be maximal. We can achieve a more compact one by resorting the core positions in this layer in a way that the gap generated by this surface point is closed (recall that a hydrophobic core is just a set of positions, with no other conditions imposed on them). Under the condition of a maximal compact layer core, this implies that every horizontal and vertical line that goes through the core in some layer must generate 2 surface points. Hence, a frame of size $a \times b$ must generate at least $2a + 2b$ surface points. Furthermore, one can conclude that an optimal frame for n points must minimize $a + b$, which is the case for a nearly quadratic frame. I.e., the best possible adaption of a quadratic frame with $a = \lceil\sqrt{n}\rceil$ and $b = \lceil\frac{n}{a}\rceil$ will have minimal surface, which used as a bound when enumerating number sequences.

Concerning the interlayer contacts for the cubic lattice, the situation is simple. Given two successive layers $x = k$ and $x = k + 1$, then every position in layer $x = k$ has exactly one neighbor position in $x = k + 1$ and vice versa. Hence, there can be at most n_k interlayer contacts between $x = k$ and $x = k + 1$ if $n_k \leq n_{k+1}$. Otherwise, if $n_k \geq n_{k+1}$, then there can be at most n_{k+1} interlayer contacts. Thus, for two successive layers with n_k and n_{k+1} core positions, there are at most

$$\min(n_i, n_{i+1})$$

many interlayer contacts.

Using an appropriate transformation, the face-centered-cubic lattice can be considered as consisting of layers of the cubic lattice, which are arranged in way such that every point in one layer has four neighbors in the next layer (see Figure 3). This implies that we can use the same definitions of layer sequences as well as layer and interlayer contacts. Furthermore, we have the same bound for the layer contacts.

For the interlayer contacts, the situation is more complicated since every position in one layer can have up to 4 neighbors in the next layer. This problem

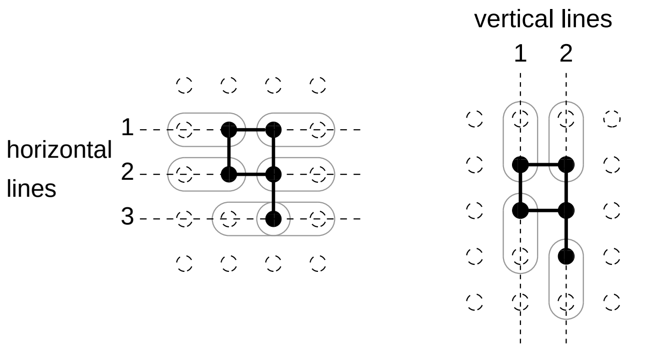


Figure 7. Horizontal and Vertical Surface. Every horizontal and vertical line through the hydrophobic core produces 2 surface points (if no surface point is buried in the core). The grey ovals include a surface point and the corresponding position of the core.

was only recently solved in [7,11]. In order to calculate the number of interlayer contacts, we have to consider in principle every possible form of the hydrophobic core in the first layer. Then, for each such form, we have to calculate how many points there are in the following layer with $1, 2, 3$ and 4 possible contacts. Formally, we define a position $\boldsymbol{p}$ in layer $x = k + 1$ to be an *i-point for the core $\mathcal{C}_k$ in layer x_k* (with $i = 1, 2, 3$ or 4) if $\boldsymbol{p}$ has i neighbors in $x = k$ that are contained in $\mathcal{C}_k$ (see Figure 8).

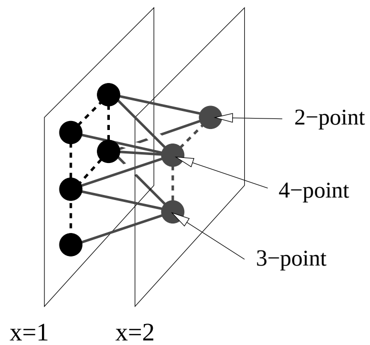


Figure 8. Definition of i-points

Of course, we don't want to enumerate all possible forms for the hydrophobic core in order to calculate a bound on the number of interlayer contacts (recall that this is intended as a first step *before* enumerating the hydrophobic cores).

But we can show that it is possible to calculate a bound on the number of i-points in layer $x = k + 1$ without knowing the *exact* form of the hydrophobic core in layer $x = k$. Instead, it is sufficient to know just the frame in layer $x = k$, which is exactly what is needed when enumerating frame sequences. This bound is given in the next theorem.

Theorem 1 ([7]). *Let $x = k$ be a layer containing n_k points in a frame with height a_k and width b_k. Furthermore, let ℓ be the number of 3-points in layer $x = k$. Then*

number 3-points $= \ell$	*number 1-points* $= \ell + 4.$
number 4-points $= n_k + 1 - a_k - b_k$	*number 2-points* $= 2a_k + 2b_k - 2\ell - 4$

It would go beyond the scope of this paper to explain this equations in detail, and the reader is referred to [7] for details. Using the above theorem, it is clear for a fixed number ℓ of 3-points, the placement of n_{k+1} in the next layer that maximizes the number of interlayer contacts is as follows. First, one fills as many 4-points as possible, then, with the remaining points, one fills the 3-points and so on. Note that this placement might *not* be optimal for the layer contacts in layer $x = k + 1$, which isn't necessary since we are calculating an upper bound. The remaining question which number of 3-points to choose. Consider the effects of going from ℓ 3-points to $(\ell + 1)$ 3-points. By the theorem, this gives us one point 3-point and one 1-point more, but we loose two 2-points. If we have enough elements in layer $x = k$ to fill even all 2-points in the case of ℓ 3-points, then we do not loose anything by going to $(\ell + 1)$ 3-points. If not, then it is better to have more 3-points. Hence, by the optimal placement strategy described above, this implies that it is best to have as much 3-points as possible. This number is also determined by the frame in $x = k$ and given in the next theorem.

Theorem 2 ([7]). *Let $x = k$ be a layer containing n_k points in a frame with height a_k and width b_k. W.l.o.g., we assume that $a_k \leq b_k$. Then the maximum number of 3-points $\ell_{\max}$ is given by*

$$\ell_{\max} = \begin{cases} 4e + r & \text{if } 4e + r < 2(a_k - 1) \\ 2(a_k - 1) & \text{else.} \end{cases}$$

where $e = \max\{s \in \mathbb{N} \mid a_k b_k - 4\frac{s(s+1)}{2} \geq n_k\}$, *and* $r = \lfloor \frac{a_k b_k - 4\frac{e(e+1)}{2} - n}{e+1} \rfloor$.

Again, the details go beyond the scope of this paper. But there is a more or less simple geometric explanation for this number which is as follows. 3-points are generated at outer diagonals of the hydrophobic core in layer $x = k$. The lengths of the diagonals depends on their distances from the corner of the frame. This is the same as the number of diagonal lines at this corner that are *not* occupied by core elements. Then e is the maximal number of diagonals that can be left unoccupied in all corners of the frame (when distributing the unoccupied diagonals equally on all corners). $r \leq 3$ is the number of times that we can add one additional unoccupied diagonal. To give an example, consider the core for

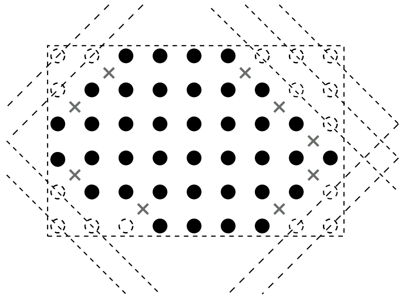


Figure 9. Diagonals and 3-points. The figure shows one specific layer of an hydrophobic core (indicated by black beads) together with it frames. $\times$ indicate 3-points in the *next* layer. One can see that in each corner, the number of 3-points is exactly the number of diagonals in that corner that do not contain a core position.

one layer in Figure 9 with $n = 38$, $a = 6$ and $b = 9$. Then e is 2. That means, in each corner we can have at least 2 diagonal lines that are not occupied by core positions. r is 1, which means that in one corner, we can add a third unoccupied diagonal.

Thus, we can calculate boundaries for every number sequences (by assuming the optimal frames for layer and interlayer contacts independently) and for frame sequences. Furthermore, we can calculate (using a dynamic programming approach, for details see [10]) optimal or near optimal frame sequences.

4 Constructing the Hydrophobic Cores

Once we have a frame sequence a_k, b_k, n_k for $k = 1 \ldots m$, one has to enumerate the possible hydrophobic cores for this frame sequence. The first step is to fix the frame positions in each layer. That is, we have finite domain variables sy_k and sy_k for the lower left corner of the frame in layer $x = k$. We can choose $sy_1 = sz_1 = 0$ for the first frame. For the remaining frames, we have to enumerate in principle all possible starting positions. But again, we can use bounds to discard combination of values for sy_k, sz_k that may not result in a maximal compact hydrophobic core.

An example of such a bound is the following. A combination is unfavorable if a frame does not completely overlap with the previous frame. Then only the part of the two frames that do overlap can generate interlayer contacts. Hence, we can use the bounds on the interlayer contacts described in the last section to calculate the number of interlayer contacts for the overlapping sub-frames.

Once we have fixed the frames (via determining their lower left corners), we start be enumerating the positions that are actually contained in the core. This can be done by inserting for every position a Boolean variable $c_{\boldsymbol{p}}$ for every position $\boldsymbol{p}$ that is in one of the fixed frames. Then

$$c_{\boldsymbol{p}} = 1 \quad \text{iff} \quad \boldsymbol{p} \text{ is in the core.}$$

Clearly, we have

$$\left(\sum_{\boldsymbol{p} \text{ is in Layer } x=k} c_{\boldsymbol{p}} \right) = n_k.$$

Since a frame is usually tightly filled, this constraint provides good propagation. Finally, we have to encode contacts by using a Boolean variable $\texttt{Contact}_{\boldsymbol{p},\boldsymbol{p}'}$ for each pair of neighbors $\boldsymbol{p}, \boldsymbol{p}'$. Then

$$\texttt{Contact}_{\boldsymbol{p},\boldsymbol{p}'} = 1 \Leftrightarrow (c_{\boldsymbol{p}} = 1 \wedge c_{\boldsymbol{p}'} = 1).$$

Counting $\texttt{Contact}_{\boldsymbol{p},\boldsymbol{p}'}$ will gives us the total number of contacts for the core.

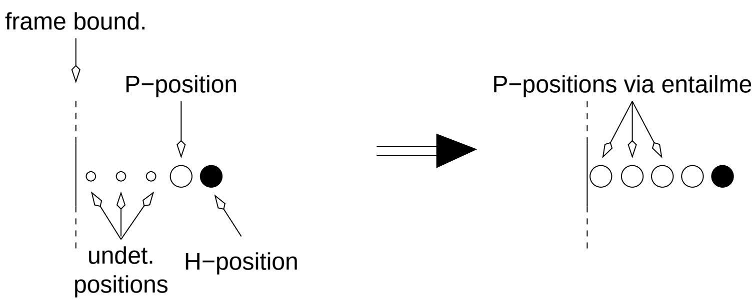


Figure 10. Example of the caveat-freeness constraint

We can improve propagation by the following consideration. Usually, hydrophobic cores do not have too many caveats. A *caveat* is a P-monomer which is part of the hydrophobic core and thus buried by H-monomers. This usually produces a non-optimal core, but must be considered in the case that the optimal cores do not correspond to a valid sequence conformation. If a frame does not contain any caveat, then we know that for any line through the frame, the H-monomers must be consecutive on this line. Now suppose we have two positions $\boldsymbol{p}$ and $\boldsymbol{p'}$ in the same frame with the property that $\boldsymbol{p}$ is the left neighbor of $\boldsymbol{p'}$, $c_{\boldsymbol{p}} = 0$ (P-position) and $c_{\boldsymbol{p}} = 1$ (H-position). Then all positions to the left of $\boldsymbol{p}$ on the line through $\boldsymbol{p}, \boldsymbol{p'}$ must be P-positions as well (see Figure 10). For a given pair of left neighbors $\boldsymbol{p} = (k, s, t)$ and $\boldsymbol{p'} = (k, s+1, t)$, this can be simply expressed by the following entailment constraint:

$$c_{\boldsymbol{p}} = 0 \wedge c_{\boldsymbol{p}'} = 1 \implies \bigwedge_{\substack{\boldsymbol{p}'' = (k,s,t) \\ \text{in frame} \\ \text{with } r < s}} c_{\boldsymbol{p}''} = 0$$

Of course, we have to introduce such a constraint for every pair of left, right or vertical neighbors. For more details, the reader is referred to [5] and [45]. If caveats are allowed, then one can enumerate them explicitly and add the constraint for the remaining positions.

5 Threading the sequence to a hydrophobic core.

The final problem is the threading of the sequence to the hydrophobic core. Once we have fixed the core, it is clear that the set of positions that can be occupied by the monomers is finite. H-monomers must take positions *inside* the hydrophobic core, and P-monomers positions *outside* the hydrophobic core. To model this mapping, we introduce variables $x_1 \ldots x_n$ for a sequence s of length n that have positions as values. This can be achieved by assigning a unique number to each position. If the i-th mom omer is an H-mom omer (i.e. $s_i = H$), then x_i can of course have only H-positions as a values, and analogously for P-monomers.

Now we have to encode the properties of conformations, which are self-avoiding walks. The self-avoidingness is nothing else than an **alldifferent** constraint on $x_1 \ldots x_n$. The walk property implies that successive monomers must occupy neighboring positions. This this nothing else then a path on a graph E, which contains as nodes all positions, and an edge for two different nodes if they are neighboring positions. Thus, we have to require in addition that $x_1 \ldots x_n$ form a path in this graph E. For this purpose, we introduce the constraint $\mathrm{Path}(x_1, \ldots, x_n)$. The question is now how to guarantee hyper-arc consistency for this constraint. By a general result of Freuder [27], arc consistency amounts to global consistency in a tree-structured network of binary constraints. The next lemma is an instance of this result.

Lemma 1. *Let $x_1, \ldots, x_n$ be variables.* $\mathrm{Path}(x_1, \ldots, x_n)$ *is hyper-arc consistent, iff for $1 \leq i \leq n-1$ all constraints* $\mathrm{Path}(x_i, x_{i+1})$ *are arc consistent.*

Due to this lemma, the hyper-arc consistency of the n-ary path constraint is reduced to the arc consistency of the set of all 2-ary path constraints $\mathrm{Path}(x_i, x_{i+1})$.

The main problem is that we will not have enough propagation if self-avoiding walks are model using the two separate and unconnected constrains $\mathrm{AllDiff}(x_1, \ldots, x_n)$ and $\mathrm{Path}(x_1, \ldots, x_n)$. To improve this we have to consider the combined constraint

$$\mathrm{SAPath}(x_1, \ldots, x_n) = \mathrm{AllDiff}(x_1, \ldots, x_n) \wedge \mathrm{Path}(x_1, \ldots, x_n).$$

Unfortunately, we are not aware of any efficient arc consistency algorithm for this combined constraint in the literature. Furthermore, it is unlikely that there exists one. It is well known that many problems involving self-avoiding walks (we use the term path here), especially counting of such walks, are intrinsically hard and there are no efficient algorithms to solve them [36].

For this reason, we have considered in [10] a relaxation of this constraint that provides better propagation but is still tractable. We have searched for a class of walks that are between arbitrary walks and self-avoiding walks. Self-avoiding means that we have the AllDiff() constraint for every variable in the walk. But what happens if we enforce this only for a subset of variables? To this end, we introduced the concept of k-avoiding walks, which are walks that are self-avoiding for every subwalk of length k (but not necessarily for the complete walk). Figure 11 shows a walk that is 4-avoiding, but neither 5-avoiding nor self-avoiding.

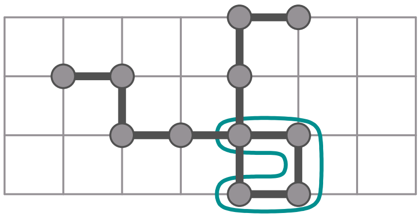


Figure 11. A walk that is not self-avoiding but 4-avoiding. Encircled is a subwalk of length 4. Every subwalk of length 4 is self-avoiding.

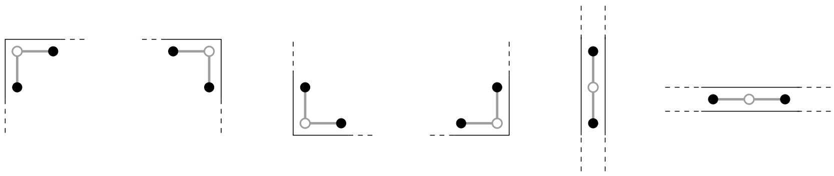


Figure 12. All possible conformations for an HPH subsequence. The black lines indicate the minimal frame boundaries, showing that the P must be within the frame enclosing the two surrounding H-monomers.

The combination of cpaths$[k](x_1,\ldots,x_n)$ (which expresses k-avoidingness) and AllDiff$(x_1,\ldots,x_n)$ provides much better propagation than the combination of Path$(x_1,\ldots,x_n)$ and AllDiff$(x_1,\ldots,x_n)$, which can be seen in the following example. The cubic lattice has the property that if we consider a HPH subsequence, then the middle P monomer must be contained in the frame that contains also the surrounding H-monomers (see Figure 12). The reason is that the only way we could have the P outside is to go back and forth, which is not allowed by the self-avoidingness condition. This property is detect via propagation of the 3-avoidingness constraint, but not using separate propagation on the constraints AllDiff$(x_1,\ldots,x_n)$ and Path$(x_1,\ldots,x_n)$.

6 Results

We have successfully applied the approach to sequences up to length 200. The hydrophobic cores are pre-calculated. Thus, when folding an HP-sequence, only the work for the threading step has to be done. Runtimes for this steps are given in Figure 13. The approach can be tested on our WWW-page

`http://www.bio.inf.uni-jena.de/Software/Prediction/prediction.cgi`

In Figure 14, a comparison with other approaches is given. Usually, no search times are given. For that reason, we have listed the maximal sequence length

Sequence	length	runtime
PPPHPHHPHHPPPHPHPPPPHPHHPPHPHHHHHPPHHPPHHHHHHPPHPP HHPPHPHPHHHHHPHHPHHHPPPHHHPHHPPHPHPPHPPPHPPHPPHPPH HHPHHHPHPPHPHHPHHHHPHPHHHPHHHPPPPPHHHHHHPPPPPPPPH HHPPHPHPPPHPHPHPHHPPHHPPPPHHHHHHPPPHHPPPPPHPPPHHPP	200	6.1 s
HPHHPPHPPPPPHHPHPHPHHPPHPPPPHHHHHHPPPHPPHHHPPHPPPP HHPPHHHPHPHHHPPHPHHPPHPHHPPPPHHPPHPPHHHHPPPPPHHHPP PPHPPPPPPHPPHHPHHHHPHHHHHHHHPPHHPPPHPHHHPHHHHHPHHP HHHPHPHHPPPPHPHHPHHHPHPPPPHPPPPPPHPHHHHHPHHPPPHPPH	200	52.6 s
HPHHHPHHPHPHPPPHHHHHPHPHPHHHHPPPHHPPPPPHHPPPPPHPHH HPPPPHPPPHHPHHPPPHPPHPPPHHHHPHHPHPPPPHHPPPHHPPHPPP HPPHHHPHHHPHPPHPHHHHPPHHPPPPHHHPHHPPHPPHHHHPPHPHPP HPHPPPPPHPHPHHHHHHHPHPHHHHHHPHHPPPHPPPPHPPPPHHHPHH	200	17.6 s

Figure 13. Example runtimes for the threading of 3 random sequences of length 200 onto a hydrophobic core of size 100 (on a Pentium IV 2.4Ghz).

handled in the references. Beside the HP model on the two-dimensional square, 3-dimensional cubic and 3-dimensional FCC lattice, we have in addition the so-called "Hetero" models. These models consider more types of interactions (not only hydrophobic/hydrophobic interactions like in the HP-model). In [39,43], the interactions where even generated by a random model resulting in one specific type of interaction for *every* pair of amino acids. This kind of models are used to make prediction about general properties of the protein folding problem.

Authors	Model	Dim.	maxlen	Algorithm	Comment
			Structure Prediction Algorithms		
Shakhnovich et al. [39]	cubic Hetero	3	27	compl. enum	fixed shape
Sali et al. [43]	(max. compact)				
Dinner et al. [25]	cubic Hetero	3	125	compl. enum	fixed shape
	(max. compact)				
Yue&Dill [48]	cubic HP	3	36	branch-and-bound	optimality proven
Yue&Dill [49]	cubic HP	3	88	branch-and-bound	optimality proven
Xia et al. [46]	tetrahedral Hetero	3	?	enumeration	restricted shape
Kaya&Chan [32]	cubic Hetero	3	55	monte carlo	
Cui et al. [22]	square HP	2	18	compl. enum	
			Approximation Algorithms		
Hart&Istrail [30]	cubic HP	3	—	approximation	$\frac{3}{8}$ of optimum
Hart&Istrail [29]	FCC-HP side chain	3	—	approximation	86% of optimum
Agarwala et al. [3]	FCC-HP	3	—	approximation	$\frac{3}{5}$ of optimum

Figure 14. Results for different lattice models by other groups.

Another distinction that has to be made when considering the sequence lengths in Figure 14 is the type of the algorithm, which is specified in the last two columns. We have all kinds of approaches including complete enumerations. The enumeration approaches either can be applied only to small sequence lengths (≤ 18), or to models, where the search space has been restricted artificially. An example here is the approach by [39,43], where only maximally compact conformations have been considered. I.e., only conformations on a $3 \times 3 \times 3$ cube have been considered, which drastically reduces the search space. This implies, that they consider only sequences of length 27, which equals the number of positions in a $3 \times 3 \times 3$ cube. For this model, one has to enumerate only all 103 346 maximally compact conformations [39]. In a later work, this was extended to $5 \times 5 \times 5$ cube having 125 positions [25].

Finally, we compare our work with the CHCC-algorithm [48,49], which is the only other approach that can find provably optimal conformations in the cubic lattice. The HP-model is not designed to generate one *single* minimal energy conformation. Instead, there are a lot of minimal energy conformations, suggesting possible topologies for a protein. The number of this minimal energy conformations for a specific sequences s is called the *degeneracy* of s. In [47], Yue et al. have given a lower bound on the degeneracy of some sequences. We have largely improved these bounds (see Figure 15).

Sequence	degeneracy found by CHCC [47]	degeneracy found by our approach
HPHHPPHHHHPHHHPPHHPPHPHHHPHPHHPPHHPPPHPPPPPPPPHH	1500×10^3	$10,677,113$
HHHHPHHPHHHHHPPHPPHHPPHPPPPPPHPPHPPPHPPHHPPHHHPH	14×10^3	$28,180$
PHPHHPHHHHHHPPHPHPPHPHHPHPHPPPHPPHHPPHHPPHPHPPHP	5×10^3	$5,090$
PHHPPPPPPHHPPPHHHPHPPHPHHPPHPPHPPHHPPHHHHHHHPPHH	188×10^3	$580,751$

Figure 15. Comparison of lower bounds on the degeneracy.

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. R. Agarwala, S. Batzoglou, V. Dancik, S. E. Decatur, S. Hannenhalli, M. Farach, S. Muthukrishnan, and S. Skiena. Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the hp model. *Journal of Computational Biology*, 4(3):275–96, 1997.
4. Richa Agarwala, Serafim Batzoglou, Vlado Dancik, Scott E. Decatur, Martin Farach, Sridhar Hannenhalli, S. Muthukrishnan, and Steven Skiena. Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP-model. *Journal of Computational Biology*, 4(2):275–296, 1997.
5. Rolf Backofen. Using constraint programming for lattice protein folding. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, and Teri E. Klein, editors, *Pacific Symposium on Biocomputing (PSB'98)*, volume 3, pages 387–398, 1998.

6. Rolf Backofen. *Optimization Techniques for the Protein Structure Prediction Problem*. Habilitationsschrift, University of Munich, 1999.
7. Rolf Backofen. An upper bound for number of contacts in the HP-model on the face-centered-cubic lattice (FCC). In Raffaele Giancarlo and David Sankoff, editors, *Proc. of the 11th Annual Symposium on Combinatorial Pattern Matching (CPM2000)*, volume 1848 of *Lecture Notes in Computer Science*, pages 277–292, Berlin, 2000. Springer–Verlag.
8. Rolf Backofen. The protein structure prediction problem: A constraint optimisation approach using a new lower bound. *Constraints*, 6:223–255, 2001.
9. Rolf Backofen and Sebastian Will. Excluding symmetries in constraint-based search. In Joxan Jaffar, editor, *Proceedings of* 5^{th} *International Conference on Principle and Practice of Constraint Programming (CP'99)*, volume 1713 of *Lecture Notes in Computer Science*, pages 73–87, Berlin, 1999. Springer–Verlag.
10. Rolf Backofen and Sebastian Will. Fast, constraint-based threading of HP-sequences to hydrophobic cores. In *Proceedings of* 7^{th} *International Conference on Principle and Practice of Constraint Programming (CP'2001)*, volume 2239 of *Lecture Notes in Computer Science*, Berlin, 2001. Springer–Verlag.
11. Rolf Backofen and Sebastian Will. Optimally compact finite sphere packings — hydrophobic cores in the FCC. In *Proc. of the 12th Annual Symposium on Combinatorial Pattern Matching (CPM2001)*, volume 2089 of *Lecture Notes in Computer Science*, Berlin, 2001. Springer–Verlag.
12. Rolf Backofen and Sebastian Will. Excluding symmetries in constraint-based search. *Constraints*, 7(3):333–349, 2002.
13. Rolf Backofen, Sebastian Will, and Erich Bornberg-Bauer. Application of constraint programming techniques for structure prediction of lattice proteins with extended alphabets. *J. Bioinformatics*, 15(3):234–242, 1999.
14. Rolf Backofen, Sebastian Will, and Peter Clote. Algorithmic approach to quantifying the hydrophobic force contribution in protein folding. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, and Teri E. Klein, editors, *Pacific Symposium on Biocomputing (PSB 2000)*, volume 5, pages 92–103, 2000.
15. Zerrin Bagci, Robert L. Jernigan, and Ivet Bahar. Residue coordination in proteins conforms to the closest packing of spheres. *Polymer*, 43:451–459, 2002.
16. Zerrin Bagci, Robert L. Jernigan, and Ivet Bahar. Residue packing in proteins: Uniform distribution on a coarse-grained scale. *J Chem Phys*, 116:2269–2276, 2002.
17. 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.
18. E Bornberg-Bauer and HS Chan. Modeling evolutionary landscapes: mutational stability, topology, and superfunnels in sequence space. *Proc Natl Acad Sci U S A*, 96(19):10689–94, 1999.
19. 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.
20. Barry Cipra. Packing challenge mastered at last. *Science*, 281:1267, 1998.
21. 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.
22. Yan Cui, Wing Hung Wong, Erich Bornberg-Bauer, and Hue Sun Chan. Recombinatoric exploration of novel folded structures: a heteropolymer-based model of protein evolutionary landscapes. *Proc. Natl. Acad. Sci. USA*, 99(2):809–14, 2002.

23. 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.
24. Ken A. Dill, Klaus M. Fiebig, and Hue Sun Chan. Cooperativity in protein-folding kinetics. *Proc. Natl. Acad. Sci. USA*, 90:1942 – 1946, 1993.
25. A. R Dinner, A Sali, and M Karplus. The folding mechanism of larger model proteins: role of native structure. *Proc. Natl. Acad. Sci. USA*, 93(16):8356–61, 1996.
26. A. Dovier, M. Burato, and F. Fogolari. Using secondary structure information for protein folding in clp(fd). In *Proc. of Workshop on Functional and Constraint Logic Programming*, volume ENTCS vol. 76, 2002.
27. Eugene C. Freuder. A sufficient condition for backtrack-free search. *Journal of the ACM*, 29:24–32, 1982.
28. S. Govindarajan and R. A. Goldstein. The foldability landscape of model proteins. *Biopolymers*, 42(4):427–438, 1997.
29. W. E. Hart and S. Istrail. Lattice and off-lattice side chain models of protein folding: linear time structure prediction better than 86 *Journal of Computational Biology*, 4(3):241–59, 1997.
30. W. E. Hart and S. C. Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal. *Journal of Computational Biology*, 3(1):53–96, 1996.
31. David A. Hinds and Michael Levitt. From structure to sequence and back again. *Journal of Molecular Biology*, 258:201–209, 1996.
32. H. Kaya and H. S. Chan. Energetic components of cooperative protein folding. *Physical Review Letters*, 85(22):4823–6, 2000.
33. Patrice Koehl and Michael Levitt. A brighter future for protein structure prediction. *Nature Structural Biology*, 6:108–111, 1999.
34. 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.
35. D. MacDonald, S. Joseph, D. L. Hunter, L. L. Moseley, N. Jan, and A. J. Guttmann. Self-avoiding walks on the simple cubic lattice. *J. Phys. A: Math. Gen.*, 33:5973–5983, 2000.
36. N. Madras and G. Slade. *The Self-Avoiding Walk*. Birkhäuser, Boston, 1993. 425 pages.
37. 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.
38. Britt H. Park and Michael Levitt. The complexity and accuracy of discrete state models of protein structure. *Journal of Molecular Biology*, 249:493–507, 1995.
39. E. I. Shakhnovich and A. M. Gutin. Enumeration of all compact conformations of copolymers with random sequence of links. *Journal Chemical Physics*, 8:5967–5971, 1990.
40. Neil J. A. Sloane. Kepler's conjecture confirmed. *Nature*, 395(6701):435–6, 1998.
41. 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.
42. Ron Unger and John Moult. Local interactions dominate folding in a simple protein model. *Journal of Molecular Biology*, 259:988–994, 1996.
43. A. Šali, E. Shakhnovich, and M. Karplus. Kinetics of protein folding. *Journal of Molecular Biology*, 235:1614–1636, 1994.

44. A. Šali, E. Shakhnovich, and M. Karplus. Kinetics of protein folding. *Journal of Molecular Biology*, 235:1614–1636, 1994.
45. Sebastian Will. Constraint-based hydrophobic core construction for protein structure prediction in the face-centered-cubic lattice. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, and Teri E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing 2002 (PSB 2002)*, Singapore, 2002. World Scientific Publishing Co. Pte. Ltd.
46. Yu Xia, Enoch S. Huang, Michael Levitt, and Ram Samudrala. Ab initio construction of protein tertiary structures using a hierarchical approach. *Journal of Molecular Biology*, 300:171 – 185, 2000.
47. K. Yue, K. M. Fiebig, P. D. Thomas, H. S. Chan, E. I. Sha khnovich, and K. A. Dill. A test of lattice protein folding algorithms. *Proc. Natl. Acad. Sci. USA*, 92(1):325–9, 1995.
48. Kaizhi Yue and Ken A. Dill. Sequence-structure relationships in proteins and copolymers. *Physical Review E*, 48(3):2267–2278, September 1993.
49. Kaizhi Yue and Ken A. Dill. Forces of tertiary structural organization in globular proteins. *Proc. Natl. Acad. Sci. USA*, 92:146 – 150, 1995.