A Constraint-Based Approach to Fast and Exact Structure Prediction in Three-Dimensional Protein Models

Rolf Backofen and Sebastian Will
({backofen,will}@inf.uni-jena.de)
Chair for Bioinformatics, Institute of Computer Science,
Friedrich-Schiller-Universität Jena, Jena Center for Bioinformatics,
Ernst-Abbe-Platz 2, D-07743 Jena, Germany

Abstract.
Simplified protein models are used for investigating general properties of proteins and principles of protein folding. Furthermore, they are suited for hierarchical approaches to protein structure prediction. A well known protein model is the HP-model of Lau and Dill [33], which models the important aspect of hydrophobicity. One can define the HP-model for various lattices, among them two-dimensional and three-dimensional ones. Here, we investigate the three-dimensional case. The main motivation for studying simplified protein models is to be able to predict model structures much more quickly and more accurately than is possible for real proteins. However, up to now there was a dilemma: the algorithmically tractable, simple protein models can not model real protein structures with good quality and introduce strong artifacts.

We present a constraint-based method that largely improves this situation. It outperforms all existing approaches for lattice protein folding in HP-models. This approach is the first one that can be applied to two three-dimensional lattices, namely the cubic lattice and the face-centered-cubic (*FCC*) lattice. Moreover, it is the only exact method for the FCC lattice. The ability to use the FCC lattice is a significant improvement over the cubic lattice. The key to our approach is the ability to compute maximally compact sets of points (used as *hydrophobic cores*), which we accomplish for the first time for the FCC lattice.

Keywords: protein structure prediction, HP-model, face-centered cubic lattice, constraint programming

1. Introduction

Proteins, the molecular machines of the cell, perform and control the essential functions in living organisms. Therefore, the investigation of their structure, the forming process of this structure, and finally their function is vital for our understanding of life.

1.1. Proteins

Viewed on the molecular level, proteins are chains composed of single building blocks. More specifically, a protein is a linear *polymer* formed by connecting *monomers*, which are called *amino acids*. An amino acid

© 2005 *Kluwer Academic Publishers. Printed in the Netherlands.*

is a molecule of the form shown in Figure 1. All amino acids share

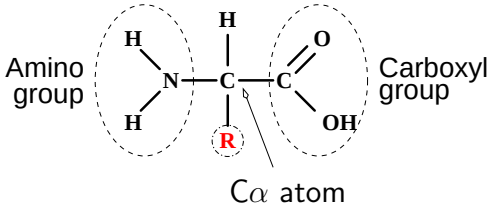


Figure 1. General structure of an amino acid.

the same general structure and differ only in the chemical group R. The central carbon atom is called the *α-carbon* (short Cα), the left group NH_2 is called the *amino group*, and the right group COOH is called *carboxy group*. In living organisms occur 20 different amino acids, which have different chemical properties. Especially, they vary in hydrophobicity, size and charge.

In a protein the amino acids are linearly arranged and thereby form a chain. The order of amino acids is called *sequence* of the protein and is specific for each protein. Note that by simple combinatorics, this allows for a huge variety of different proteins.

In order to form a chain, every two amino acids are connected via a *peptide bond*, where the carboxy group of the first amino acid reacts with the amino group of the second. The result of connecting two amino acids is a molecule as shown in Figure 2.

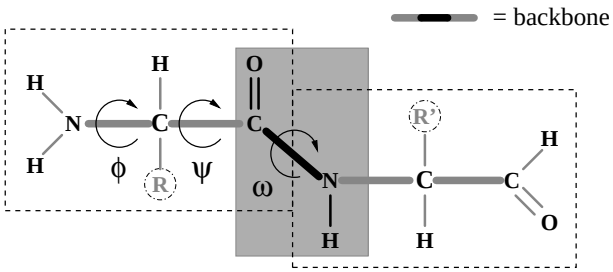


Figure 2. Two amino acids connected by a peptide bond.

The peptide bond itself, which is indicated with a grey rectangle in Figure 2, is planar, which means that there is no free rotation around this bond.[1] There is more flexibility for rotation around the N-Cα-bond

[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 usually occurs only in combination with the amino acid Proline.

(called the ϕ-angle), and around the Cα–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.

Using this freedom of rotation, the protein can form a huge variety of different three-dimensional structures. Due to thermodynamics, some of them are energetically more favorable than others, i.e. these conformations have lower energy and therefore are more stable than the other ones. As of yet, the details of this energy function are not completely clear and a matter of intensive research. It is commonly assumed that for a natural protein, there is one distinguished conformation that has minimal energy, which is uniquely determined by the sequence of amino acids. For this reason, one speaks of the *native structure* of a protein denoting this distinguished conformation. *Protein folding* denotes the conformational search of a protein, which culminates in finding the native conformation. This term is distinguished from the term *protein structure prediction*, which denotes the computation of the native structure from the sequence. In this paper, we deal with structure prediction and are not concerned with kinetic aspects of the protein folding process.

Finally, the native structure and sequence determine the function of the protein, due to the general mode of operation of proteins.

1.2. Structure Prediction

Here, we discuss to which extent the three-dimensional structure of a protein can be computationally inferred from its sequence alone.

To assess the value of computational structure prediction, it should be mentioned that experimental structure determination is still difficult and time consuming[2]. In consequence, current biochemical methods have difficulty to keep pace with the rapid growth of the number of known protein sequences.

Therefore, protein structure prediction is one of the most important problems of computational biology, which is however still unsolved in general. We specify this problem in the following way. Given a protein by its sequence of amino acids, what is its native structure? Since as already discussed the native structure is the structure with minimal energy, protein structure prediction is reasonably modeled as an optimization problem. The NP-completeness of this problem has been proven for many different formal, in general simplified, protein models including lattice and off-lattice models. [15, 20]

[2] Here, the structure is determined by using either X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy.

These results strongly suggest that the protein folding problem for real proteins is NP-hard. Therefore, it is unlikely that a general, efficient algorithm for solving this problem can be given. In fact, one is not able to answer this question definitively, since too little is known about the general principles of protein folding.

Knowing these general principles not only would improve our capabilities to predict a structure of a protein, but it is also of paramount importance for rational drug design, where one faces the difficulty to design proteins that have a unique and stable native structure.

1.3. Simplified Models of Proteins

We will discuss mainly the most important sub-class of simplified protein models, although parts of this sub-section apply to simplified models in general. The models in this class are called *lattice models*. The simplifications commonly used in this class of models are:

- each monomer is modeled by only one point,
- the positions of the monomers are restricted to lattice positions,
- all monomers have equal size,
- all bonds are of equal length, and
- a simplified energy function is used.

For the aim of structure prediction, gaining insight into the relationship between sequence and structure of proteins is of utmost importance. *Simplified protein models*, also known as low-resolution or coarse-grained protein models, were proposed to study this relationship. Furthermore, they can be used to tackle structure prediction directly.

For both areas of application, the following consideration motivates the use of simplified models. Since simplified models model only some aspects of the protein's structure and energy, it is in general easier to compute optimal structures for the model's proteins than for real proteins. The same applies for solving related problems like the design of sequences that fold to a given structure.

For the latter application area of direct structure prediction, simplified models have been successfully used by several groups in the international contest "Critical Assessment of Techniques for Protein Structure Prediction" (CASP, see the meeting review of CASP3 [32]). There, the models are used in hierarchical approaches for protein folding [46] (see also Figure 3). In general, these approaches use simplified models in a filter step as follows. Given a protein sequence, first a

set of good structures in the simplified model is generated (e.g. 10 000 structures). In subsequent steps, these candidate structures are fine-tuned using more computationally involved methods. Usually, these methods incorporate biological knowledge and simulation of protein folding on full atomic detail (i.e. molecular dynamics simulation).

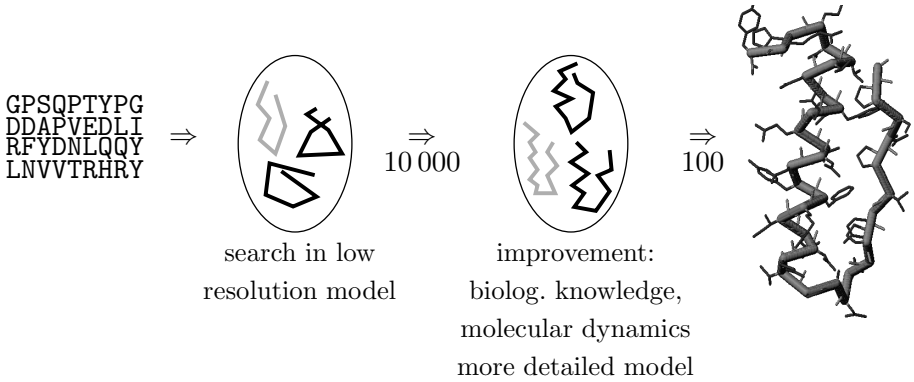


Figure 3. Hierarchical Approach to Protein Structure Prediction

Apart from their use in structure prediction and regarding the relationship of sequence and structure, lattice models have become a major tool for investigating general properties of protein folding. They constitute a genotype (protein sequence) to phenotype (protein conformation) mapping that can be handled using computational methods. An interesting application of such a mapping is the exploration of evolutionary processes. An example is [17], where the arrangement of sequences in neutral nets is shown. A *neutral net* consists of sequences that all code for the same structure and that are connected by point mutations. In neutral nets there is a prototype sequence, which encodes the common structure in the most stable way. For the sequences of the net, this stability decreases with their distance to the prototype sequence. The assumed universality of this principle feeds the hypothesis of superfunnels in the sequence space with direct implications to protein design. Another exciting question in this context is whether one can switch between two different neutral nets using only a small number of amino acid substitutions. This could help to explain how the diversity of protein conformations evolved, which is observed 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 have 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, 22]), the second aspect is underrepresented. By and large, mainly two different heuristic search approaches are used in the literature. The first approach is an ad hoc restriction of the search space to compact or quasi-compact conformations. A good example is [34], 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 a 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 too strongly and then may exclude minimal conformations. The second approach is to use stochastic sampling as performed by Monte Carlo methods with or without simulated annealing or genetic algorithms. Here, the degree of optimality for the best conformations and the quality of the sampling cannot be determined by state of the art methods.[3]

1.4. Contribution of the Paper: A Constraint-based approach

We discuss protein structure prediction in an important class of lattice models, which is know as the class of HP-models. In this work, we introduce a constraint-based approach that greatly improves over all existing approaches in HP-model lattice protein folding.

Originally, the term HP-model has been introduced by Lau and Dill in [33] to denote a two-dimensional square lattice model with an energy function that is simplified as strong as possible. In this model, the 20 letter alphabet of amino acids is reduced to an alphabet of two letters, namely H and P. H represents *hydrophobic* amino acids, whereas P represents *polar* or hydrophilic amino acids. The energy function for the HP-model is given by the matrix of Figure 4(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 the Euclidian distance of their

[3] When using stochastic local search methods, the partition function of the ensemble (which is needed for a precise statement) remains in general unknown.

a)

	H	P
H	-1	0
P	0	0

b)

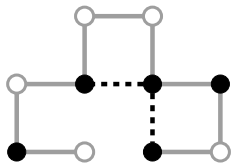


Figure 4. Energy matrix and sample conformation for the HP-model

positions is 1 and they are not connected via a bond.[4] A conformation with *minimal energy* (also called *optimal conformation*) is just a conformation with the maximal number of contacts between H-monomers. NP-completeness of the structure prediction problem has been shown even for the HP-model [15, 20].

A sample conformation for the sequence PHPHPPHHPH in the two-dimensional square lattice with energy -2 is shown in Figure 4(b). The white beads represent P, the black ones H monomers. The two contacts are indicated via dashed lines.

Originally, the HP-model was defined for the two-dimensional square lattice. However, the extension to other lattices is straightforward. For example, an HP-model using the face-centered cubic lattice (FCC) is investigated in [3].

As we will explain in the following, our structure prediction approach outperforms other approaches in several ways, namely flexibility, completeness and efficiency.

Concerning flexibility, our method is the only one that works for two different important three-dimensional lattices. These are the cubic lattice and the face-centered-cubic lattice. The cubic lattice is the most intensively studied three-dimensional lattice. However, the ability of the cubic lattice to approximate real protein conformations is poor. Furthermore, as [3] pointed out, there is a parity problem in the cubic lattice. This means that every two monomers with chain positions of the same parity cannot form a contact. Nevertheless, we support the cubic lattice due to its wide-spread use in literature and for comparability of our method to other structure prediction approaches.

In contrast, the FCC overcomes the discussed drawbacks of the cubic lattice. It lacks the parity problem and models real protein conformations with good quality (see [38], where it was shown, that the FCC lattice can model protein conformations with coordinate root mean square deviation of 1.78 Å, whereas the cubic lattice achieves a devia-

[4] Note that this second condition can be dropped without changing the optimization problem, since it adds only a constant number of contacts. Actually, we will do this later.

tion of 2.84 Å there). Recently, [13, 14] have shown that neighborhood 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 [41, 19] (see Figure 5 for an illustration of the FCC lattice).

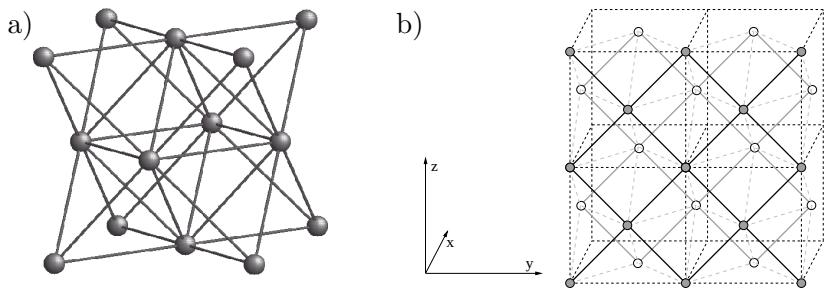


Figure 5. a) The unit cell of the FCC. b) A cut-out of two layers of the face-centered-cubic lattice (FCC). The layers can be seen as two square lattices, which are shifted such that every position in the first layer has contacts to 4 positions in the second layer and vice versa (shown as dashed lines). Since the FCC lattice is continued by stacked layers of the square lattice that are shifted each time, every position of the FCC has twelve neighbors (four within the same layer, four to the previous layer, and four to the next layer).

Concerning completeness, we find optimal structures that can not be computed by all other comparable approaches. The fact that the HP-model is degenerated[5] allows for a direct comparison of completeness. The degeneracy (i.e. the number of optimal structures) of a sequence can be computed by enumerating all optimal conformations. Only for the cubic lattice, there is one other method that claims to completely enumerate optimal conformations [48] in a large class of conformations *and* proves their optimality. In [48], Yue and Dill give a lower bound for the number of such conformations for some sequences, by enumerating as many structures as their algorithm can find. For these sequences we can significantly improve their lower bound, which shows that the CHCC method is incomplete. Note that an incomplete algorithm can not only miss optimal conformations, but even fail to determine the optimal energy for structures of a sequence.

Concerning efficiency, we have successfully applied our algorithm to sequences up to length 200 in the face-centered cubic lattice (FCC). For several sequences of length 200, we found a minimal energy con-

[5] In a degenerated model there are usually many optimal conformations for one sequence, instead of only one optimal conformation as it is assumed for real proteins.

formation *and* proved its optimality. For the FCC, so far there existed only heuristic algorithms (for an example of a genetic algorithm for arbitrary Bravais lattices see [12]). Usually, these algorithms are applied to sequence of length of at most 80 (where they usually find only a low but *not* minimal energy conformation). Since the search space for conformations in the cubic lattice 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.[6]

1.5. Related Work

A discussion of simplified protein models and their benefit can be found in [22]. There is a number 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, 24, 43, 30, 27, 2, 37]. Most of them use heuristic methods, ranging from Monte-Carlo simulated annealing (e.g. [35, 24]) to genetic algorithms (e.g. [43]), purely heuristic methods like hydrophobic zipper [23] and the chain growth algorithm [16], as well as complete enumeration (often restricted to subset of all conformations, e.g. [44, 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 finding minimal energy conformations in the cubic HP-Model [48]. The algorithm is called CHCC for "**C**onstraint **H**ydrophobic **C**ore **C**onstruction". Note that albeit its name, the approach does not use constraint-based methods. This method 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 approximation algorithms [29, 3] for different lattice models. Despite their indisputable merits, their approximation ratio is still too weak to be used in practice.

An approach of predicting structures using a FCC lattice-model and also constraint-technology is the work of [25]. There, secondary structure annotations, i.e. which amino acids form α-helices and β-sheets, are employed to restrict the search space for finding solutions in a lattice models. Despite some superficial similarities to our approach, both methods have different application areas in mind. The

[6] The number 4.5^n has been estimated for the cubic lattice [36]; for the FCC, we are not aware of a good estimation of the number of conformations. However, due to the increased degrees of freedom in the FCC lattice this number is certainly higher than the number of conformations in the cubic lattice.

aim of their work is to find not necessarily optimal solutions in a rather detailed protein model, i.e. they compute only an approximation to the prediction problem. Although their method cannot guarantee the quality of found solutions, the method is slow and is only applicable to short proteins. In contrast, we trade some detail of the model for predicting structures, which are provably optimal and can be computed much faster. Therefore, our approach is suited for applications in large scale. In particular, we bring to mind statistical studies and generating topological models in hierarchical structure prediction.

2. Overview of the Constraint-Based Approach

In Sub-section 2.1, we describe a straightforward constraint model for the protein structure prediction problem. The simple model is then followed by an overview of our more sophisticated approach in the next sub-section.

2.1. A First Constraint Model

The following ideas are applicable to the HP-model in arbitrary lattices. However for simplicity, we introduce the formal model for the cubic lattice only. The description for other lattices, e.g. the face-centered-cubic lattice, is analogous, especially since every lattice has an integral representation, which allows using the finite domain constraint system (for details of the application to the FCC lattice see [5]).

A sequence is a word of the alphabet $\{H, P\}$. With s_i we denote the ith element of a sequence s. We call two lattice points $\vec{p}$ and $\vec{p}'$ *(lattice) neighbors* if and only if $\vec{p}$ and $\vec{p}'$ have the minimal euclidian distance between any two lattice points, which is 1 in the cubic lattice.

A *conformation* (also called a *structure*) c of a sequence s is a function $c : [1..|s|] \to \mathbb{Z}^3$ such that

$$\text{for all } 1 \leq i < |s|,\ c(i) \text{ and } c(i+1) \text{ are neighbors} \qquad \text{(bonds)}$$
$$\text{and for all } i \neq j,\ c(i) \neq c(j). \qquad \text{(excluded-volume)}$$

By the bond constraint, two successive monomers have to be lattice neighbors. The second condition, the excluded-volume constraint, claims that a conformation is self-avoiding, i.e. it does not overlap itself. Thus, the two constraints formulate the common requirements for a conformation in the HP-model.

Given a conformation c of a sequence s, the *number of HH-contacts* $\text{HHContacts}(s, c)$ in c is defined as the number of pairs (i, j) with $1 \leq$

$i < j \leq |s|$, where

$$s_i = H,\ s_j = H,\ \text{and } c(i) \text{ and } c(j) \text{ are neighbors.}$$

The energy of c is commonly defined in terms of the contacts as

$$-\text{HHContacts}(s, c)\ +\ h,$$

where h is the number of pairs $(i, i+1)$, for $1 \leq i < |s|$, where $s_i = H$ and $s_{i+1} = H$. Note that since the number h is constant for a given sequence, maximizing the number of HH-contacts is equivalent to minimizing the energy.

Now, we can use constraints for defining the set of possible structures of a given sequence. When we give this constraint encoding in the following, please note that all constraints can be expressed using the constraint system over finite integer domains and furthermore Boolean and reified constraints, which are defined in the following.

A *reified constraint* is a constraint $\mathtt{x} \leftrightarrow \phi$, where ϕ denotes an allowed constraint. Semantically, by the constraint $\mathtt{x} \leftrightarrow \phi$ the truth of ϕ is reified in the value of the Boolean variable $\mathtt{x}$, i.e. $\mathtt{x}$ is 1 if ϕ holds and 0 otherwise. Furthermore, we use entailment constraints of the form $\phi \rightarrow \psi$ with the semantic ϕ implies ψ. For handling both constraints, the solver needs to implement the entailment test. A constraint is called *entailed* if ϕ is satisfied by every valuation that satisfies our constraint problem. The constraint is *disentailed* if its negation is entailed. The reified constraint $\mathtt{x} \leftrightarrow \phi$ is then handled by determining the value of $\mathtt{x}$, when the solver derives that ϕ is entailed or disentailed. $\phi \rightarrow \psi$ is handled by imposing the constraint ψ, when ϕ is known to be entailed.

Our constraint model can be directly implemented in the language Oz [42], since this programming system supports Boolean and finite domain constraints, the entailment test, and a programmable search module.

For the actual constraint model, 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. The domains are also finite, since we can restrict the possible values of these variables to $[\,1\,..\,2|s|\,]$.[7] This is expressed by introducing the constraints

$$\mathtt{X}_i \in [\,1\,..\,2|s|\,] \wedge \mathtt{Y}_i \in [\,1\,..\,2|s|\,] \wedge \mathtt{Z}_i \in [\,1\,..\,2|s|\,]$$

[7] We even could have used the domain $[1..n]$. However, the domain $[\,1\,..\,2|s|\,]$ 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.

for every $1 \leq i \leq n$. The excluded-volume constraint is just given for $i \neq j$ by

$$(\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i) \neq (\mathtt{X}_j, \mathtt{Y}_j, \mathtt{Z}_j).^{8}$$

For expressing that two successive monomers are lattice neighbors, we introduce for every monomer i with $1 \leq i < |s|$ three finite domain variables $\mathtt{Xnext}_i$, $\mathtt{Ynext}_i$ and $\mathtt{Znext}_i$. Then, we can express the bond constraint by

$$\begin{array}{rclrcl} \mathtt{Xnext}_i & = & |\mathtt{X}_i - \mathtt{X}_{i+1}| & \mathtt{Znext}_i & = & |\mathtt{Z}_i - \mathtt{Z}_{i+1}| \\ \mathtt{Ynext}_i & = & |\mathtt{Y}_i - \mathtt{Y}_{i+1}| & \mathtt{Xnext}_i + \mathtt{Ynext}_i + \mathtt{Znext}_i & = & 1. \end{array}$$

The previously described constraints span the space of all possible conformations. Every valuation of $\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i$ satisfying these constraints is an *admissible* conformation for the sequence s, i.e. a self-avoiding walk of s.

In order to search for conformations with maximal number of HH-contacts, we add constraints for counting HH-contacts. Then, one can optimize the number of HH-contacts by enumerating the variables $\mathtt{X}_i$, $\mathtt{Y}_i$ and $\mathtt{Z}_i$. For the counting, one introduces a Boolean variable $\mathtt{Contact}_{i,j}$ for each $1 \leq i < j \leq |s|$ that is 1 if i and j have a contact in every conformation that is compatible with the valuations of $\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i$ and 0 otherwise. Then, for $1 \leq i < j \leq |s|$, we introduce new FD-variables $\mathtt{Xdiff}_{i,j}$, $\mathtt{Ydiff}_{i,j}$, and $\mathtt{Zdiff}_{i,j}$ and constrain them by

$$\begin{array}{rclrcl} \mathtt{Xdiff}_{i,j} & = & |\mathtt{X}_i - \mathtt{X}_j| & \mathtt{Zdiff}_{i,j} & = & |\mathtt{Z}_i - \mathtt{Z}_j| \\ \mathtt{Ydiff}_{i,j} & = & |\mathtt{Y}_i - \mathtt{Y}_j| & \mathtt{Contact}_{i,j} & \in & \{0,1\} \end{array}$$

$$\mathtt{Contact}_{i,j} \leftrightarrow (\mathtt{Xdiff}_{i,j} + \mathtt{Ydiff}_{i,j} + \mathtt{Zdiff}_{i,j} = 1). \tag{1}$$

Finally, the variable $\mathtt{HHContacts}$ counts the number of contacts between H-monomers, and is defined by

$$\mathtt{HHContacts} = \sum_{\substack{i<j \\ s_i=s_j=H}} \mathtt{Contact}_{i,j}. \tag{2}$$

Now for computing a structure with maximal number of HH-contacts, we apply constraint-based enumeration of the variables $\mathtt{X}_i$, $\mathtt{Y}_i$, and $\mathtt{Z}_i$. There, we use a branch-and-bound scheme in order to optimize $\mathtt{HHContacts}$.

The previous formulation is general enough to be easily upgraded to other lattices or more complex energy functions. However, when

[8] This cannot be directly encoded in Oz [42], but we reduce these constraints to difference constraints on integers.

we use the previously described approach alone, the search space will be restricted only poorly. We want to give some explanation for the unsatisfying performance. For the previous approach, an efficient constraint solver is too weak in detecting whether a partial structure has the potential to achieve sufficiently many contacts. For improving this ability, one needs to introduce additional constraints to get bounds on the number of contacts. However, we do not see how to define good bounds using the previous approach. Furthermore, in order to apply branch-and-bound optimization, one needs a search heuristic that prefers low-energy conformations. Again, it seems unlikely to find such a heuristic.

Consequently, for strongly improving over the previous constraint model, we can not simply enhance this model. Instead, we develop a completely new approach, which is described in the following section.

2.2. New Structure Prediction Approach

The whole approach relies on the notion of hydrophobic cores. The *hydrophobic core* of a protein structure is the set of positions of the hydrophobic amino acids. One observes that native structures tend to have very compact hydrophobic cores. In fact, the energy of a structure in the HP-models is completely dictated by the number of contacts in its hydrophobic core C, which is defined by

$$\text{contacts}(\mathcal{C}) = \frac{1}{2}|\{(\vec{p}, \vec{p}') \mid \vec{p}, \vec{p} \in \mathcal{C} \wedge \vec{p} \text{ and } \vec{p} \text{ are neighbors}\}|.$$

Thus, the native structure for an HP-sequence s has a hydrophobic core which is as compact as possible for s.

Let n_{H} denote the number of H-monomers in s. Then usually, the number of contacts in the hydrophobic core of the native structure of s is very close to the maximal number of contacts between n_{H} lattice points. This observation allows for a new structure prediction strategy.

For predicting an optimal structure of s, one tries to find structures of s that have one of the most compact point sets as hydrophobic core. If this fails, one goes on to less compact point sets until one finds a structure of s. If we find a first structure for s by this strategy, we have already proved that it is an optimal structure, since we search exhaustively and in each iteration decrease the number of contacts of the point sets only by one.

In order to follow this new strategy in our approach, we need to solve two separate problems. First, we need to generate compact point sets (also called cores). This core construction is a complex combinatorial problem of its own and is therefore preceeded by a preparation step,

which reduces this complexity by computing core descriptions (later called *frame sequences*). Second, we need to thread an HP-sequence to a core, such that the core becomes the hydrophobic core of the resulting structure.

The resulting approach of predicting optimal structures for an HP-sequence s is depicted in Figure 6. This figure shows, that we finally end up with three steps *bounding*, *core construction*, and *threading*. Furthermore it shows the possible iteration, which relaxes the number of contacts in the cores by one, if threading fails.

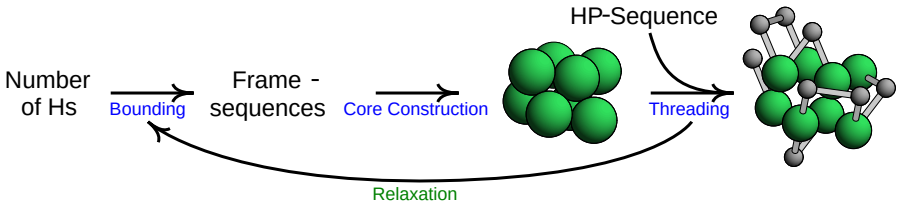


Figure 6. Schematic overview of the structure prediction approach.

An immediate advantage of our approach is that the cores can be constructed independently of an actual sequence and can be stored for later use in structure prediction. That is, the first two steps bounding and core construction can be regarded pre-computation steps and do not influence the run-time for actual structure prediction, where we access already computed and stored cores.

The most challenging part of the method is the construction of compact point sets, where we want to enumerate the point sets of a given size that have maximally many or (slightly) less contacts.

A direct enumeration of such point sets seems to be not feasible, since we are faced with a huge search space. However, if we get information about the shape of such point sets, we can enumerate the cores due to the restriction of the search space.

The first idea for restricting possible cores is to define the surrounding cuboid that contains the hydrophobic core. If one has a very tight cuboid, then the hydrophobic core in this cuboid must be rather compact. This claim obviously holds for the cubic lattice and is also of some use for the FCC.

However, this approach is not fine-grained enough for the FCC as well as for sub-optimal hydrophobic cores in the cubic lattice. Therefore, we introduce a tighter bound on the number of contacts, which is obtained by splitting the lattice into layers. Here, a *layer* is a plane that

is 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 7). Please note that the exact position of the frames is *not* part of the frame sequence. For each frame sequence, there is an upper bound on the number of contacts in every hydrophobic core that has this frame sequence. It is possible to efficiently compute the set of frame sequences for n monomers that all have at least a bound c. Then, each core of size n with c contacts must have one of these frame sequences.

Using this information, we perform a constraint-based search for enumerating all hydrophobic cores of size n with c contacts. This search is strongly restricted by the frame sequences for n H-monomers with a bound of at least c contacts.

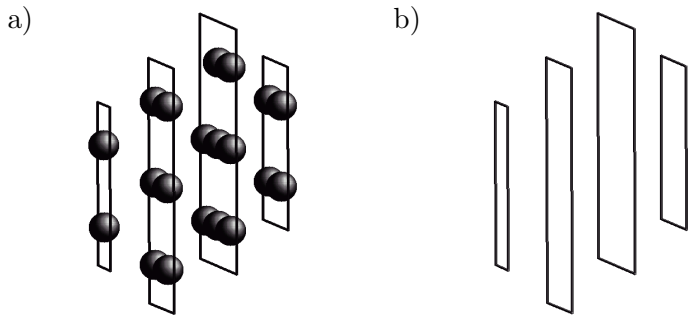


Figure 7. Hydrophobic cores and frame sequences. a) a hydrophobic core with frames. b) the corresponding frame sequence. a_i is the width and b_i is the height of the frame in the i-th layer. n_i gives the number of H-monomers in this layer.

Note that hydrophobic cores for structure prediction were first used in the CHCC algorithm of Yue and Dill [47]. CHCC works only for the cubic lattice and not for the FCC lattice. For the cubic lattice, both methods, CHCC and our approach, bound the number of contacts. However, CHCC only computes the cuboids that surround the cores with a certain number of contacts. By further considering the distribution of elements to layers, our method yields even stronger constraints on the compact cores than CHCC. This is especially important for the enumeration of sub-optimal cores. For FCC, there is no method that, as it is done for the cubic lattice by CHCC, directly computes the cuboid-analogous, sphere-like shapes that surround the compact cores.

Threading of an actual HP-sequence to a core is illustrated in Figure 8. Threading can be modeled as a constraint satisfaction problem

and is then solved by constraint-based search. As well as the search that is used for core construction, this search profits very much from the use of a symmetry breaking scheme that is described in [8, 11]. There, we introduced the first general method for the breaking of symmetries in constraint-based search, which now has become a standard method.

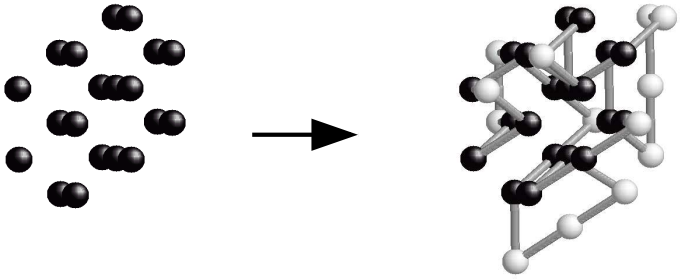


Figure 8. Threading of the HP-sequence 'HHPPPPHHHHPHHHPHHHPPHHPHH-PPPHHHPH' to a hydrophobic core in the FCC lattice. The figure shows the core (on the left) and the resulting structure (on the right), where H-monomers are shown as black beads and P-monomers as white ones.

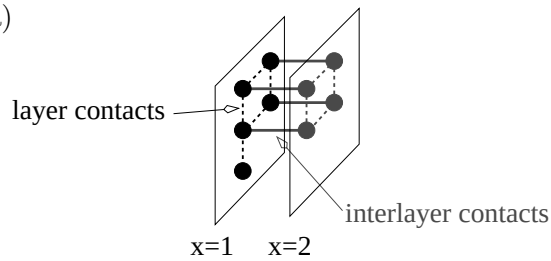
In the next sections, we will discuss the single steps of our prediction method. The highly lattice-specific bounds on contacts will be described in a section of their own. After the description of the bounds, we will explain the construction of the hydrophobic cores, and finally describe the threading method.

3. An Upper Bound for Frame Sequences

As prerequisite for the enumeration of hydrophobic cores, we investigate the problem of generating the set of all frame sequences for a given number of points with a bound of at least c contacts.

The first step is to define the upper bound on contacts for a given frame sequence $(a_1, b_1, n_1) \ldots (a_l, b_l, n_l)$, which is discussed separately for the cubic lattice and the FCC lattice. We will start with the less complex case of the cubic lattice. Note that for the cubic lattice, there exists a previous bound on contacts by Yue and Dill [47]. However, we present our new bound for two reasons. First, we can improve over their bound by investigating the distribution of H-monomers to layers. Second, the bound is instructive for understanding the more intricate case of the FCC lattice, since both bounds share a similar structure.

a)



b)

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

Figure 9. a) Layer and Interlayer Contacts b) Corresponding Frame Sequence

3.1. A Bound for Frame Sequences in the Cubic Lattice

In Figure 9a), a hydrophobic core for the cubic lattice is shown, where we explicitly mark its two layers. Figure 9b) gives the corresponding frame sequence. All contacts between positions in the same layer are called *layer contacts*. All contacts between positions in successive layers are called *interlayer contacts*. The upper bound on the number of contacts in any core satisfying the given frame sequence is defined as the sum of separate bounds for the number of layer *and* interlayer contacts.

In order to bound the layer contacts in the cubic lattice, we employ the concept of surface, which was used by [47] before. If we define a *surface pair of a core* $\mathcal{C}$ as a pair $(\vec{p}, \vec{p}')$ of lattice points that have unit distance, where $\vec{p} \in \mathcal{C}$ and $\vec{p}' \notin \mathcal{C}$. Then, the *surface of a core* $\mathcal{C}$ is the number of its surface pairs.

Now, imagine a single layer $x = k$ of the lattice that intersects the core. Then, we define the *layer surface* of a hydrophobic core $\mathcal{C}$ in layer $x = k$ as the number of surface pairs of $\mathcal{C}$, where both positions are in this layer.

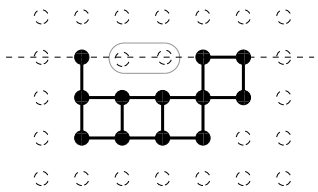


Figure 10. The figure shows a layer of the cubic (or FCC) lattice that has one cavity (grey oval). In general, a *cavity of a core (resp. layer)* is a non empty set of positions on one line that are not in the core, but are framed by two core positions on this line.

We assume that there are n core elements in this layer $x = k$ and these positions are contained in a minimal rectangle of size $a \times b$, which is called the *frame of the layer*. Then, the surface and layer contacts

are related via the equation

$$4 \cdot n = 2 \cdot \text{Contacts} + \text{Surface} \tag{3}$$

since every core element $\vec{p}$ has four neighbors in the same layer and each of these neighbors $\vec{p}'$ is either in the core or not. In the former case, $(\vec{p}, \vec{p}')$ (but also $(\vec{p}', \vec{p})$) contributes to the number of contacts by $\frac{1}{2}$, in the latter case $(\vec{p}, \vec{p}')$ contributes to the surface by 1.

Hence, minimizing the surface maximizes the number of contacts. Yue and Dill [47] observed for the cubic lattice that minimizing the surface is easier than directly maximizing the number of layer contacts. In particular, if there are no cavities in the core (see Figure 10 for the meaning of cavity), then the layer surface is given by $2 \cdot (a+b)$ (compare Figure 11). Thus, $2 \cdot (a+b)$ is a lower bound on the surface. Using this in Equation 3 yields an upper bound on the number of contacts from a, b, and n.

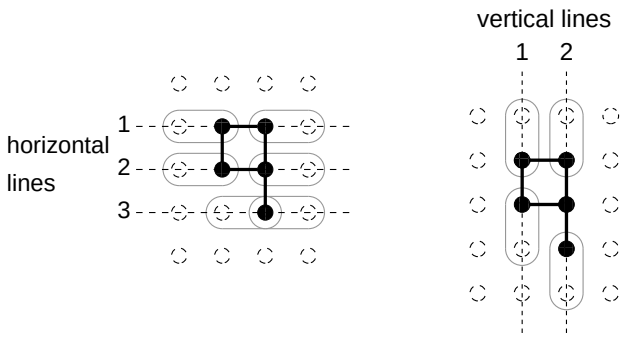


Figure 11. Horizontal and Vertical Surface. Every horizontal and vertical line through the hydrophobic core produces at least two surface pairs (or exactly two surface pairs, if there are no cavities in the core). The grey ovals mark the pairs of surface points and corresponding core positions.

For the cubic lattice, there is a straightforward upper bound on the number of interlayer contacts. Given two successive layers $x = k$ and $x = k + 1$, every position in layer $x = k$ has exactly one neighbor position in $x = k + 1$ and vice versa. Hence, there can not be more than n_k or n_{k+1} interlayer contacts between $x = k$ and $x = k + 1$. Thus, for two successive layers with n_k and n_{k+1} core positions, there are at most $\min(n_i, n_{i+1})$ interlayer contacts.

3.2. A Bound for Frame Sequence in the FCC lattice

Our key to bounds for the FCC lattice is to partition the face-centered cubic lattice into layers that each form a square lattice (as in the cubic

lattice). In the FCC, these layers are arranged in a way that every point in one layer has four neighbors in the next layer (see Figure 5). Note that due to the partitioning, the definitions from the cubic lattice of layer sequences, layers, and interlayer contacts apply also for the FCC lattice. Furthermore, we can use the same bound for the layer contacts as in the case of the cubic lattice.

For the interlayer contacts in the the face-centered-cubic lattice, the situation is more intricate than in case of the cubic lattice since every position in a layer can form a contact with up to four neighbors in the next layer. The problem of bounding these contacts was only recently solved in [5, 10].

The key problem for bounding the total number of interlayer contacts is the bounding of interlayer contacts between two successive layers $x = k$ and $x = k + 1$. Obviously, it is not feasible to search through all possible pairs of layers that satisfy the parameters a_k, b_k, n_k and $a_{k+1}, b_{k+1}, n_{k+1}$ in order to obtain a tight bound.

However, imagine that we know the distribution of monomers in the layer $x = k$. Then, we can count how many points in the layer $x = k + 1$ form $1, 2, 3$ and 4 contacts to the first layer. Formally, we define a position $\vec{p}$ in layer $x = k + 1$ to be an *i-point for the core $\mathcal{C}$ in layer $x = k$* (with $i = 1, 2, 3$ or 4) if $\vec{p}$ has i neighbors that are contained in layer $x = k$ and $\mathcal{C}$ (see Figure 12). We get a bound on the number of interlayer contacts by distributing the n_{k+1} elements of the second layer to these i-points. There, we fill the positions greedily, i.e. starting with 4-points and continuing with decreasing i.

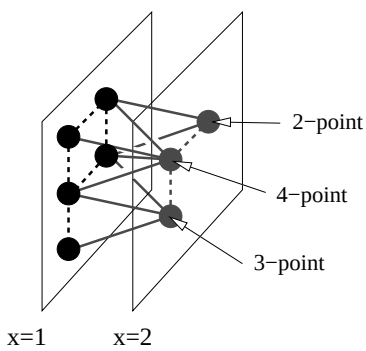


Figure 12. Definition of i-points.

In [10], we argue that there is a relation between the frame $a \times b$, the number of elements n, and the numbers of i-points of a layer. Only three further parameters of the layer, denoted by $m_{\text{no}}, m_{\text{nc}}$, and m_x, are sufficient to determine the numbers of i-points exactly. Here, the

additional parameters characterize a layer in the following way. See Figure 13 for an illustration of the terms. The points of a layer can be grouped into lines that each include all layer points with the same y-coordinate. We call two successive lines in a layer overlapping if there are two points of the two lines with the same z-coordinate. Similarly, we call two successive lines connected if there are two points of the two lines, where their z-coordinates differ by 0 or 1. An x-step is a set of four neighbored lattice points that form a square, where exactly three of them are elements of the layer. Then, m_{no} denotes the number of pairs of non overlapping successive lines, m_{nc} is the number of pairs of not connected successive lines, and m_x refers to the number of x-steps of a layer.

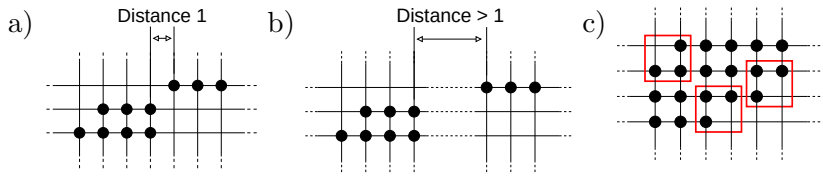


Figure 13. Additional terms for more detailed description of frames. a) The first and second line (from the top) do not overlap, but are connected. The second and third line are connected and overlap. The number of pairs of successive lines that do not overlap is denoted by m_{no}, i.e. here $m_{\mathrm{no}} = 1$. b) The first and second line do not overlap and are not connected. The number of such pairs of successive lines is given by m_{nc}, i.e. here $m_{\mathrm{nc}} = 1$. c) A layer with three x-steps, which are marked by squares. The number of x-steps is denoted by m_x, i.e. here $m_x = 3$. An x-step is characterized by three points in the layer which form a square together with a fourth point that is not element of the layer.

The numbers of i-points, written $\#i$, for $i = 1, \ldots, 4$ are given by

$$\begin{aligned}
\#4 &= n - \frac{1}{2}s + 1 + m_{\mathrm{no}} \\
\#3 &= m_x - 2(m_{\mathrm{no}} - m_{\mathrm{nc}}) \\
\#2 &= s - 4 - 2\#3 - 3m_{\mathrm{no}} - m_{\mathrm{nc}} \\
\#1 &= \#3 + 2m_{\mathrm{no}} + 2m_{\mathrm{nc}} + 4.
\end{aligned}$$

First, note that always $m_{\mathrm{no}} \geq m_{\mathrm{nc}}$ holds since an overlap of successive lines implies their connection. Furthermore, in order to obtain an upper bound we can replace m_{nc} by its bound m_{no} for the aim of eliminating one parameter. In this way, we get

$$\begin{aligned}
\#3' &= m_x \\
\#2' &= s - 4 - 2\#3' - 4m_{\mathrm{no}} \\
\#1' &= \#3' + 4m_{\mathrm{no}} + 4.
\end{aligned}$$

Thereby, the number of 4-points is unaffected and $\#3'$ and $\#1'$ possibly overestimate the numbers $\#3$ and $\#1$, which is compensated by an underestimation of $\#2$ by $\#2'$.

Now, the bound on the interlayer contacts of the layer with frame $a \times b$, n elements, m_{no} non-overlapping lines, and m_x x-steps to a further layer with n' elements is obtained by distributing the n' elements to positions that form as many interlayer contacts as possible. Formally, if we define four auxiliary variables by

$$\begin{aligned}
b_4 &= min(n', \#4) \\
b_3 &= min(n' - b_4, \#3') \\
b_2 &= min(n' - b_4 - b_3, \#2') \\
b_1 &= min(n' - b_4 - b_3 - b_2, \#1'),
\end{aligned}$$

then the bound is simply given by $4 \cdot b_4 + 3 \cdot b_3 + 2 \cdot b_2 + b_1$.

Finally, if every other parameter is fixed, this bound is maximized if the number of x-steps is maximal. In [5], a closed formula is given for the maximal number of x-steps in overlapping layers. [10] extends this to layers with non-overlapping lines by giving a recursion formula, which is efficiently evaluated using dynamic programming.

This allows us to compute a bound by only enumerating the possible values of m_{no} instead of enumerating the exponentially many possible layers.

3.3. Generating Frame Sequence Sets

Now, we discuss the generation of a set of frame sequences for cores with given size n and a bound of at least c contacts. This generation is discussed for both lattices uniformly.

We start by computing a bound $\mathrm{B_C}(n, n_1, a_1, b_1)$ on the number of contacts in cores of size n and a first layer $x = 1$ that has n_1 elements and the frame $a_1 \times b_1$. This can be done efficiently for all n up to some upper limit and all n_1, a_1, b_1 at the same time using a dynamic programming (DP) algorithm. This algorithm fills a four-dimensional matrix for evaluating the recursion of Equation (4) (Figure 14 provides an illustration). We define two functions $\mathrm{B_{LC}}$ and $\mathrm{B_{ILC}}$, which denote the lattice specific bounds as they are described above. The function $\mathrm{B_{LC}}(n_1, a_1, b_1)$ (resp. $\mathrm{B_{ILC}}(n_1, a_1, b_1, n_2, a_2, b_2)$) denotes the upper bound of the contacts on layer contacts for layers with parameters n_1, a_1, b_1 (resp. interlayer contacts between the two layers with parameters n_1, a_1, b_1 and n_2, a_2, b_2).

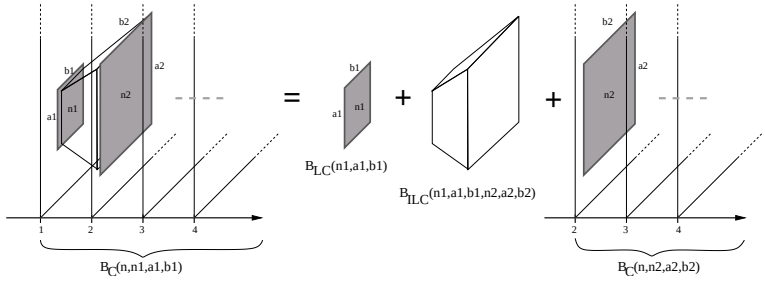


Figure 14. Illustration of Equation 4. Recursively, the equation reduces the bound for cores with first layer $x = 1$ to a bound of cores with first layer $x = 2$. The recursion equation abstracts from the continuation of the core in layers $x = 3, 4, \ldots,$ which allows for efficient evaluation.

$$\mathrm{B_C}(n, n_1, a_1, b_1) = \max_{n_2, a_2, b_2} \left(\begin{array}{ll} & \mathrm{B_{LC}}(n_1, a_1, b_1) \\ + & \mathrm{B_{ILC}}(n_1, a_1, b_1, n_2, a_2, b_2) \\ + & \mathrm{B_C}(n - n_1, n_2, a_2, b_2) \end{array} \right) \quad (4)$$

The frame sequence sets are generated by trace-back through the resulting four-dimensional matrix. Since we are interested in the frame sequences with at least a bound of c contacts, these sequences are not necessarily optimal. Note that we also generate these sub-optimal frame sequences from the DP-matrix, which is done by tolerating a limited deviation from the optimal path when computing the trace-back.

The interested reader will find a more complete discussion of the generation of bounded frame sequences in [10] and further detail in [7].

4. Constructing the Hydrophobic Cores

In order to construct the hydrophobic cores of size n with at least c contacts, we use the corresponding complete set of frame sequences to restrict a constraint-based search.

Given the set of frame sequences, we know that each core must have one of these frame sequences. Otherwise it could not form the required number of contacts c, due to our bound of the previous section. For the cubic lattice, it is furthermore straightforward that each core must have one of the frame sequences in every possible layer decomposition, i.e. either a decomposition along the x-axis into x-layers, one into y-layers, or one into z-layers.

Now, for gaining the maximal information from the sequences also for FCC, one has to understand how the x-, y-, and z-layers are oriented

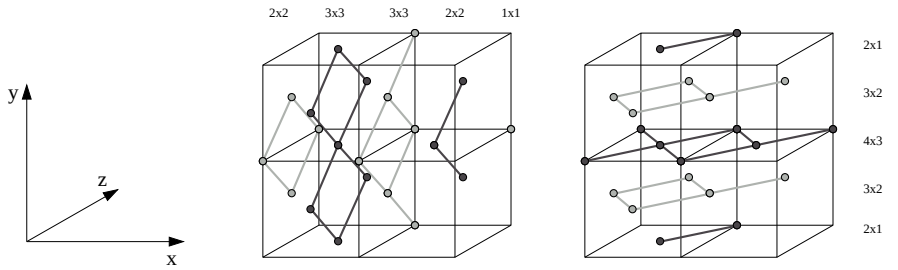


Figure 15. Two representations of a single hydrophobic core in the FCC lattice. The cores are embedded in a cubic structure to emphasize the building principle of this lattice as face-centered-cubic. We have marked the x-layers (resp. y-layers) by showing their layer contacts. The use of light and dark ink emphasizes the layer structure.

to each other in the FCC lattice. Figure 15 illustrates that the layers of different dimensions x,y, or z are orthogonal to each other as in the cubic lattice. However, in contrast to the cubic lattice, they can be imagined as being rotated by 45°. Due to this arrangement, for the FCC lattice we can apply the same constraint as for the cubic lattice, i.e. that one of the frame sequences must be satisfied in either dimension.

The previous constraint is introduced in the form of a disjunction. Note that since general constructive disjunction is computationally expensive, we have to use a specialized approach to extract information out of this special disjunction and improve the propagation by generating element constraints.

In order to enumerate the cores, we start by fixing the length of the frame sequence in every dimension. Since we search for connected cores, the lengths immediately tells us the dimensions of a minimal surrounding cuboid that contains all points of the core and furthermore contains no empty layers.[9]

Notably, not all combinations of sequence lengths can be satisfied, which is however hard to detect at this stage of the search. Therefore in case of the FCC lattice, we simply perform a complete enumeration of frame sequence lengths. For the cubic lattice, the CHCC algorithm of [47] provides means to restrict the dimensions of the cuboid by an upper bound on the number of contacts. Here, we combine our own bound with a bound that we derive from this part of the CHCC algorithm. However, we had to enhance CHCC in order to handle the case of cores with sub-optimally many contacts completely.

[9] Note that we construct only connected cores; unconnected cores can still be composed from connected ones.

As soon as the surrounding cuboid is fixed, we introduce boolean variables $\texttt{CorePos}_{\vec{p}}$ for every lattice point in the cuboid. These variables tell whether the point belongs to the core. The variables $\texttt{CorePos}_{\vec{p}}$ are constrained to the (still partially known) frame dimensions and elements in each layer. Furthermore, the variables are constrained to the number of contacts in each layer and in total. For this aim, for each pair of points $\vec{p}$ and $\vec{q}$, a boolean variable $\texttt{Contact}_{\vec{p},\vec{q}}$ is introduced. For example, the constraint for the total number of contacts is then easily expressed as

$$\sum \texttt{Contact}_{\vec{p},\vec{q}} >= c.$$

Before enumerating the boolean variables directly, it is advantageous to introduce boolean variables for each line along the lattice vectors that intersects the cuboid. These variables tell whether the points of the line are occupied. Then, these variables are enumerated first. Finally, constraints that count the overlapping and connection of lines, as well as constraints that relate the surface of layers and the whole core, improve the constraint propagation during the search. Further details of this approach are given in [45] where we discuss the core construction problem for the FCC lattice.

5. Threading the sequence to a hydrophobic core

The final problem is the threading of the sequence to a hydrophobic core (see Figure 8), which yields the structures where the H-monomers build the given hydrophobic core. This is discussed independently of the actual lattice. We define a *self-avoiding walk* as a sequence of lattice positions where successive positions are lattice neighbors and no position occurs twice. Shortly, the *threading problem* asks for a self-avoiding walk where all H-monomers are placed on core positions.

When given an HP-sequence s of length n and a core $\mathcal{C}$, we model the problem as CSP using the finite domain constraint system. We start by introducing finite domain variables $\texttt{X}_1 \ldots \texttt{X}_n$. The values of these variables are the positions of the corresponding monomers in the FCC lattice. Therefore, a valuation can encode a protein structure in our model. First note that these variables have indeed finite domains. This is a consequence of the fact that the positions of H-monomers are in the finite core $\mathcal{C}$ and the P-monomers are connected to the H-monomers. Regarding the implementation, note that we can still use a standard finite domain constraint system with integer domains if we assign a unique number to each position.

The restriction of the H-monomers to core positions is now simply expressed by unary constraints[10]

$$\mathtt{X}_i \in \mathcal{C} \text{ for } 1 \leq i \leq n,\ s_i = H. \tag{5}$$

The self-avoiding property of a conformation means that all positions of monomers have to be different, which is directly expressed by an *all-different* constraint on $\mathtt{X}_1 \ldots \mathtt{X}_n$. Hence we introduce

$$\text{AllDiff}(\mathtt{X}_1, \ldots, \mathtt{X}_n). \tag{6}$$

Technically, we use the constraint of difference a la Regin [39] for the H-monomers, which ensures hyper-arc consistency[11], and a weaker propagating constraint for the P-monomers. Thus, we use the computationally expensive complete all-different constraint only where it propagates most efficiently.

The walk property claims that successive monomers must occupy neighboring positions. To ensure this property, we introduce the constraint $\text{Walk}(\mathtt{X}_1, \ldots, \mathtt{X}_n)$. We investigate now how we can guarantee hyper-arc consistency for this constraint. By a general result of Freuder [26], 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.* $\text{Walk}(X_1, \ldots, X_n)$ *is hyper-arc consistent if and only if for* $1 \leq i \leq n-1$ *all constraints* $\text{Walk}(X_i, X_{i+1})$ *are arc consistent.*

Due to this lemma, the hyper-arc consistency of the n-ary walk constraint is reduced to the arc consistency of the set of all 2-ary walk constraints $\text{Walk}(\mathtt{X}_i, \mathtt{X}_{i+1})$.

We observed that the propagation is still rather weak if self-avoiding walks are modeled using the two separate constraints $\text{AllDiff}(\mathtt{X}_1, \ldots, \mathtt{X}_n)$ and $\text{Walk}(\mathtt{X}_1, \ldots, \mathtt{X}_n)$ that communicate only over the domains of the variables. To improve the propagation, we discuss the combined constraint

$$\text{SAWalk}(\mathtt{X}_1, \ldots, \mathtt{X}_n) = \text{AllDiff}(\mathtt{X}_1, \ldots, \mathtt{X}_n) \wedge \text{Walk}(\mathtt{X}_1, \ldots, \mathtt{X}_n).$$

Unfortunately, we are not aware of any efficient arc consistency algorithm for this combined constraint in the literature. Furthermore, it

[10] A *unary constraint* depends on only one variable.

[11] A constraint is hyper-arc consistent, if all domain values of its variables are supported by an assignment of all variables to their domain values that satisfies the constraint. Arc consistency denotes the special case of hyper-arc consistency for 2-ary constraints.

is unlikely that there exists one. It is well known that many problems involving self-avoiding walks, especially counting of such walks, are intrinsically hard and there are no efficient algorithms to solve them [36].

For this reason, we have investigated in [9] a relaxation of the self-avoiding walk constraint that provides better propagation but is still tractable. For variables $\mathtt{X}_1, \ldots, \mathtt{X}_n$, SAWalk$(\mathtt{X}_1, \ldots, \mathtt{X}_n)$ introduces the condition that all variables have different values, which makes constraint propagation hard. Obviously, we can reduce the complexity if we enforce the all-different condition only for smaller subsets of the variables. We observed that a reasonable choice is to guarantee the self-avoiding property only for each set of k successive variables. In order to formalize this, we introduce the concept of k-avoiding walks, which are walks that are self-avoiding for every sub-walk of length k (but not necessarily for the complete walk). Figure 16 shows a walk that is 4-avoiding, but neither 5-avoiding nor self-avoiding. The constraint Walk$[k](\mathtt{X}_1, \ldots, \mathtt{X}_n)$ constrains the variables $\mathtt{X}_1, \ldots, \mathtt{X}_n$ to form a k-avoiding walk.

In our constraint model, we can now combine the all-different constraint with the k-avoiding walk constraint instead of combining it with the walk constraint. Note that both constraint formulations are equivalent, i.e. they have the same set of solutions. However, the interplay of Walk$[k](\mathtt{X}_1, \ldots, \mathtt{X}_n)$ and AllDiff$(\mathtt{X}_1, \ldots, \mathtt{X}_n)$ provides much better propagation than the one of the two constraints Walk$(\mathtt{X}_1, \ldots, \mathtt{X}_n)$ and AllDiff$(\mathtt{X}_1, \ldots, \mathtt{X}_n)$. This can be seen in the following example. The cubic lattice has the property that if we consider an HPH subsequence, then the middle P monomer must be contained in the frame that contains also the surrounding H-monomers (see Figure 17). 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-avoiding condition. This property is detect via propagation of the 3-avoiding constraint, but not

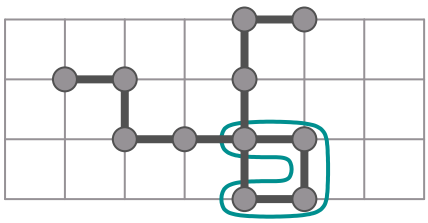


Figure 16. A walk that is not self-avoiding but 4-avoiding. Encircled is a sub-walk of length 4. Every sub-walk of length 4 is self-avoiding.

using separate propagation on the constraints $\mathrm{AllDiff}(\mathtt{X}_1, \ldots, \mathtt{X}_n)$ and $\mathrm{Walk}(\mathtt{X}_1, \ldots, \mathtt{X}_n)$.

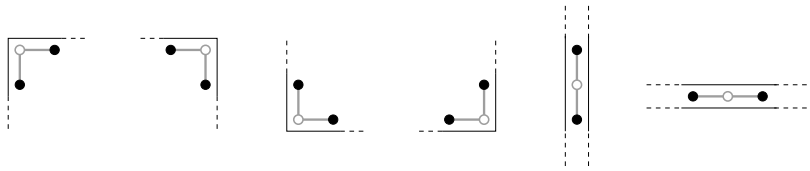


Figure 17. All possible conformations for an HPH subsequence. The black lines indicate the minimal frame boundaries. This illustrates that the P must be within the frame enclosing the two surrounding H-monomers.

Due to a generalization of Lemma 1, we render $\mathrm{Walk}[k](\mathtt{X}_1, \ldots, \mathtt{X}_n)$ hyper-arc consistent by computing the hyper-arc consistency of the $n - k + 1$ many constraints $\mathrm{SAWalk}(\mathtt{X}_i, \ldots, \mathtt{X}_{i+k-1})$ for all $1 \leq i \leq n-k+1$. Notably, in the special case $k = 3$, the consistency of the 3-ary constraints $\mathrm{SAWalk}(\mathtt{X}_i, \mathtt{X}_{i+1}, \mathtt{X}_{i+2})$ is computed in only time $O(2b \cdot d)$, where d is the domain size of the variable $\mathtt{X}_{i+1}$ and b is the constant number of neighbors of each lattice point, namely $b = 6$ for the cubic lattice and $b = 12$ for the FCC.

6. Results

We have successfully applied the approach for predicting optimal structures for sequences up to length 200. When folding an HP-sequence,

Table I. Example runtimes for the threading of three random sequences of length 200 onto a hydrophobic core of size 100 in the FCC lattice (on a Pentium 4 at 2.4GHz).

Sequence	length	runtime
PPPHPHHPHHPPPHPHPPPPHPHHPPHPHHHHHPPHHPPHHHHHHPPHPP HHPPHPHPHHHHHPHHPHHHPPPHHHPHHPPHPHPPHPPPHPPHPPHPPH HHPHHHPHPPHPHHPHHHHPHPHHHPHHHPPPPPPHHHHHHPPPPPPPPH HHPPHPHPPPHPHPHPHHPPHHPPPPHHHHHHPPPHHPPPPPHPPPHHPP	200	6.1 s
HPHHPPHPPPPPHHPHPHPHHPPHPPPPHHHHHHPPPHPPHHHPPHPPPP HHPPHHHPHPHHHPPHPHHPPHPHHPPPPHHPPHPPHHHHPPPPPHHHPP PPHPPPPPPHPPHHPHHHHPHHHHHHHHPPHHPPPHPHHHPHHHHHPHHP HHHPHPHHPPPPHPHHPHHHPHPPPPHPPPPPPPHPHHHHHPHHPPPHPPH	200	52.6 s
HPHHHPHHPHPHPPPHHHHHPHPHPHHHHPPPHHPPPPPPHHPPPPHPHH HPPPPHPPPHHPHHPPPHPPHPPPHHHHPHHPHPPPPHHPPPHHPPHPPP HPPHHHPHHHPHPPHPHHHHPPHHPPPPHHHPHHPPHPPHHHHPPHPHPP HPHPPPPPHPHPHHHHHHHPHPHHHHHHPHHPPPPHPPPPHPPPHHHPHH	200	17.6 s

only the work for the threading step has to be done, since the hydrophobic cores are pre-calculated. Runtimes for the threading step are given in Table I. The approach can be tested on our WWW-page:

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

In Table II, a comparison with other approaches is given. Usually, no search times are given in the literature. For that reason, we have listed the maximal sequence length handled in the references. Beside the HP model on the two-dimensional square, the three-dimensional cubic and the three-dimensional FCC lattice, there are models that distinguish a larger number of amino acids and thus more types of interactions than just the hydrophobic/hydrophobic interactions in the HP-model. These models are commonly called "Hetero" models. In [40, 44], the interactions were even generated by a random model resulting in one specific type of interaction for *every* pair of amino acids. Models of this kind are used to make prediction about general properties of the protein folding problem.

Table II. Results for different lattice models by other groups.

Authors	Model	Dim.	maxlen	Algorithm	Comment
	Structure Prediction Algorithms				
Shakhnovich et al. [40] and Sali et al. [44]	cubic Hetero (max. compact)	3	27	compl. enum	fixed shape
Yue&Dill [47]	cubic HP	3	36	b&b	proves optimum
Yue&Dill [48]	cubic HP	3	88	b&b	proves optimum
Xia et al. [46]	tetrahedral Hetero	3	?	enumeration	restricted shape
Kaya&Chan [31]	cubic Hetero	3	55	monte carlo	
Cui et al. [21]	square HP	2	18	compl. enum	
	Approximation Algorithms				
Hart&Istrail [29]	cubic HP	3	—	approx.	$\frac{3}{8}$ of optimum
Hart&Istrail [28]	FCC-HP side chain	3	—	approx.	86% of optimum
Agarwala et al. [3]	FCC-HP	3	—	approx.	$\frac{3}{5}$ of optimum

When comparing the sequence lengths in Table II, it is important to keep in mind the type of the algorithm, which is specified in the last two columns. The table lists references for all kinds of approaches ranging from complete enumeration to stochastic optimization methods. Complete enumeration is necessarily restricted. 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 [40, 44] where only maximally compact conformations are enumerated for computing the optimal structure.

Namely, only conformations on a $3 \times 3 \times 3$ cube are taken into account, which drastically reduces the search space. In consequence, 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 [18].

Finally, we compare our work with the CHCC-algorithm [47, 48], which is the only other approach that can find provably optimal conformations in the cubic lattice HP-model and in the same time prove their optimality. The HP-model is not designed to generate one *single* minimal energy conformation for each sequence. Instead, commonly there are a lot of minimal energy conformations. The number of this minimal energy conformations for a specific sequences s is called the *degeneracy* of s. In [49], Yue et al. have given a lower bound on the degeneracy of some sequences. We have largely improved these bounds (see Table III). Only for one sequence (third entry of Table III), CHCC found approximately[12] as many optimal structures as we could find. Note that we also tested the validity of our results by an independent program, which checked the generated optimal structures for uniqueness (where symmetric structures are considered equal) and correct energy.

Table III. Comparison of lower bounds on the degeneracy from our algorithm to bounds that are computed using CHCC [49]. The degeneracies for CHCC are cited as given in the reference.

Sequence	Degeneracy	
	CHCC [49]	our approach
HPHHPPHHHHPHHHPPHHPPHPHH HPHPHHPPHHPPPHPPPPPPPPHH	1.5×10^6	$10,677,113$
HHHHPHHPHHHHHPPHPPHHPPHP PPPPPHPPHPPPHPPHHPPHHHPH	14×10^3	$28,180$
PHPHHPHHHHHHPPHPHPPHPHHP HPHPPPHPPHHPPHHPPHPHPPHP	5×10^3	$5,090$
PHHPPPPPPHHPPPHHHPHPPHPH HPPHPPHPPHHPPHHHHHHHPPHH	188×10^3	$580,751$

[12] Since numbers for CHCC were not given with full precision in the reference, we can not know whether the numbers match exactly or only approximately.

7. Conclusion

We introduced the first approach to exact protein structure prediction in a FCC lattice model. For the cubic lattice, we improved strongly over previous approaches. Due to the advances in efficiency it is for the first time possible to use three-dimensional exact protein structure prediction in a large scale and for realistic sizes. For studies of the sequence structure relation, it was up to now not feasible to make use of three-dimensional models that are not artificially restricted. The completeness of our approach is crucial for many of those studies. The flexibility of the approach allows applying it to the cubic and the face-centered cubic lattice. Finally using the FCC lattice, which closely approximates real protein structures, the approach could be suited as a filter step in hierarchical protein structure prediction.

Acknowledgments

We like to thank the anonymous reviewers for their valuable comments, which helped to improve the quality of this article.

References

1. Abkevich, V. I., A. M. Gutin, and E. I. Shakhnovich (1995). Impact of local and non-local interactions on thermodynamics and kinetics of protein folding. *Journal of Molecular Biology* **252**, 460–471.
2. Abkevich, V. I., A. M. Gutin, and E. I. Shakhnovich (1997). Computer simulations of prebiotic evolution. In: *Proc. of the Pacific Symposium on Biocomputing (PSB'97)*. pp. 27–38.
3. Agarwala, R., S. Batzoglou, V. Dancik, S. E. Decatur, M. Farach, S. Hannenhalli, S. Muthukrishnan, and S. Skiena (1997). Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP-model. *Journal of Computational Biology* **4**(2), 275–296.
4. Backofen, R. (1999). Optimization techniques for the protein structure prediction problem. Habilitationsschrift. University of Munich.
5. Backofen, R. (2000). An upper bound for number of contacts in the HP-model on the face-centered-cubic lattice (FCC). In: *Proc. of the 11th Annual Symposium on Combinatorial Pattern Matching (CPM2000)*. Vol. 1848 of *Lecture Notes in Computer Science*. Berlin. pp. 277–292.
6. Backofen, R. (2001). The protein structure prediction problem: a constraint optimization approach using a new lower bound. *Constraints* **6**, 223–255.
7. Backofen, R. (2003). A polynomial time upper bound for the number of contacts in the HP-model on the face-centered-cubic lattice (FCC). *Journal of Discrete Algorithms*.
8. Backofen, R. and S. Will (1999). Excluding symmetries in constraint-based search. In: *Proc. of* 5th *International Conference on Principle and Practice of Constraint Programming (CP'99)*. Vol. 1713 of *Lecture Notes in Computer Science*. Berlin. pp. 73–87.
9. Backofen, R. and S. Will (2001). Fast, constraint-based threading of HP-sequences to hydrophobic cores. In: *Proc. of* 7th *International Conference on Principle and Practice of Constraint Programming (CP'2001)*. Vol. 2239 of *Lecture Notes in Computer Science*. Berlin.
10. Backofen, R. and S. Will (2001). Optimally compact finite sphere packings — hydrophobic cores in the FCC. In: *Proc. of the 12th Annual Symposium on Combinatorial Pattern Matching (CPM 2001)*. Vol. 2089 of *Lecture Notes in Computer Science*. Berlin.
11. Backofen, R. and S. Will (2002). Excluding symmetries in constraint-based search. *Constraints* **7**(3), 333–349.
12. Backofen, R., S. Will, and P. Clote (2000). Algorithmic approach to quantifying the hydrophobic force contribution in protein folding. In: *Pacific Symposium on Biocomputing (PSB 2000)*. Vol. 5. pp. 92–103.
13. Bagci, Z., R. L. Jernigan, and I. Bahar (2002). Residue coordination in proteins conforms to the closest packing of spheres. *Polymer* **43**, 451–459.
14. Bagci, Z., R. L. Jernigan, and I. Bahar (2002). Residue packing in proteins: uniform distribution on a coarse-grained scale. *Journal of Chemical Physics* **116**, 2269–2276.
15. Berger, B. and T. Leighton (1998). Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. In: *Proc. of the Second Annual International Conference on Research in Computational Molecular Biology (RECOMB'98)*. pp. 30–39.

16. Bornberg-Bauer, E. (1997). Chain growth algorithms for HP-type lattice proteins. In: *Proc. of the First Annual International Conference on Research in Computational Molecular Biology (RECOMB'97)*. pp. 47–55.
17. Bornberg-Bauer, E. and H. S. Chan (1999). Modeling evolutionary landscapes: mutational stability, topology, and superfunnels in sequence space. *PNAS* **96**(19), 10689–10694.
18. Chan, H. S. and K. A. Dill (1990). The Effects of Internal Constraints on the Configurations of Chain Molecules. *Journal of Chemical Physics* **92**, 3118–3135
19. Cipra, B. (1998). Packing challenge mastered at last. *Science* **281**, 1267.
20. Crescenzi, P., D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis. (1998). On the complexity of protein folding. In: *Proc. of STOC*. pp. 61–62.
21. Cui, Y., W. H. Wong, E. Bornberg-Bauer, and H. S. Chan (2002). Recombinatoric exploration of novel folded structures: a heteropolymer-based model of protein evolutionary landscapes. *PNAS* **99**(2), 809–814.
22. Dill, K. A., S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas, and H. S. Chan (1995). Principles of protein folding – a perspective of simple exact models. *Protein Science* **4**, 561–602.
23. Dill, K. A., K. M. Fiebig, and H. S. Chan (1993). Cooperativity in protein-folding kinetics. *PNAS* **90**, 1942 – 1946.
24. Dinner, A. R., A. Šali, and M. Karplus (1996). The folding mechanism of larger model proteins: role of native structure. *PNAS* **93**(16), 8356–8361.
25. Dovier, A., M. Burato, and F. Fogolari (2002). Using secondary structure information for protein folding in CLP(FD). In: *Proc. of Workshop on Functional and Constraint Logic Programming*. Vol. ENTCS vol. 76.
26. Freuder, E. C. (1982). A sufficient condition for backtrack-free search. *Journal of the ACM* **29**, 24–32.
27. Govindarajan, S. and R. A. Goldstein (1997). The foldability landscape of model proteins. *Biopolymers* **42**(4), 427–438.
28. Hart, W. E. and S. Istrail (1997). Lattice and off-lattice side chain models of protein folding: linear time structure prediction better than 86% of optimal. *Journal of Computational Biology* **4**(3), 241–259.
29. Hart, W. E. and S. C. Istrail (1996). Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal. *Journal of Computational Biology* **3**(1), 53–96.
30. Hinds, D. A. and M. Levitt (1996). From structure to sequence and back again. *Journal of Molecular Biology* **258**, 201–209.
31. Kaya, H. and H. S. Chan (2000). Energetic components of cooperative protein folding. *Physical Review Letters* **85**(22), 4823–4826.
32. Koehl, P. and M. Levitt (1999). A brighter future for protein structure prediction. *Nature Structural Biology* **6**, 108–111.
33. Lau, K. F. and K. A. Dill (1989). A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules* **22**, 3986 – 3997.
34. Li, H. and R. Helling, C. Tang and N. Wingreen (1996). Emergence of preferred structures in a simple model of protein folding. *Science* **273**, 666–669.
35. MacDonald, D., S. Joseph, D. L. Hunter, L. L. Moseley, N. Jan, and A. J. Guttmann (2000). Self-avoiding walks on the simple cubic lattice. *Journal of Physics A: Math. Gen.* **33**, 5973–5983.
36. Madras, N. and G. Slade (1993). *The self-avoiding walk*. Birkhäuser, Boston. 425 pages.

37. Ortiz, A. R., A. Kolinski, and J. Skolnick (1998). Combined multiple sequence reduced protein model approach to predict the tertiary structure of small proteins. In: *Proc. of the Pacific Symposium on Biocomputing 1998 (PSB'98)*. Vol. 3. pp. 375–386.
38. Park, B. H. and M. Levitt (1995). The complexity and accuracy of discrete state models of protein structure. *Journal of Molecular Biology* **249**, 493–507.
39. Regin, J.-C. (1994). A filtering algorithm for constraints of difference. In: *Proc. of the 12th National Conference of the American Association for Artificial Intelligence*. pp. 362–367.
40. Shakhnovich, E. I. and A. M. Gutin (1990). Enumeration of all compact conformations of copolymers with random sequence of links. *Journal of Chemical Physics* **8**, 5967–5971.
41. Sloane, N. J. A. (1998). Kepler's conjecture confirmed. *Nature* **395**(6701), 435–436.
42. Smolka, G. (1995). The Oz programming model. In: *Computer Science Today*. Lecture Notes in Computer Science. Berlin: Springer-Verlag. pp. 324–343.
43. Unger, R. and J. Moult (1996). Local interactions dominate folding in a simple protein model. *Journal of Molecular Biology* **259**, 988–994.
44. Šali, A., E. Shakhnovich, and M. Karplus (1994). Kinetics of protein folding. *Journal of Molecular Biology* **235**, 1614–1636.
45. Will, S. (2002). Constraint-based hydrophobic core construction for protein structure prediction in the face-centered-cubic lattice. In: *Proc. of the Pacific Symposium on Biocomputing 2002 (PSB 2002)*. Singapore.
46. Xia, Y., E. S. Huang, M. Levitt, and R. Samudrala (2000). Ab initio construction of protein tertiary structures using a hierarchical approach. *Journal of Molecular Biology* **300**, 171 – 185.
47. Yue, K. and K. A. Dill (1993). Sequence-structure relationships in proteins and copolymers. *Physical Review E* **48**(3), 2267–2278.
48. Yue, K. and K. A. Dill (1995). Forces of tertiary structural organization in globular proteins. *PNAS* **92**, 146 – 150.
49. Yue, K., K. M. Fiebig, P. D. Thomas, H. S. Chan, E. I. Shakhnovich, and K. A. Dill (1995). A test of lattice protein folding algorithms. *PNAS* **92**(1), 325–329.