Constraints, 6, 223–255, 2001

The Protein Structure Prediction Problem: A Constraint Optimization Approach using a New Lower Bound

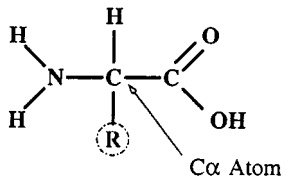
ROLF BACKOFEN
Institut für Informatik, Ludwig-Maximilians-Universität München, Oettingenstraße 67, D-80538 München

Abstract. The protein structure prediction problem is one of the most (if not *the most*) important problem in computational biology. This problem consists of finding the conformation of a protein with minimal energy. Because of the complexity of this problem, simplified models like Dill's HP-lattice model [15], [16] have become a major tool for investigating general properties of protein folding. Even for this simplified model, the structure prediction problem has been shown to be NP-complete [5], [7]. We describe a constraint formulation of the HP-model structure prediction problem, and present the basic constraints and search strategy. Of course, the simple formulation would not lead to an efficient algorithm. We therefore describe redundant constraints to prune the search tree. Furthermore, we need bounding function for the energy of an HP-protein. We introduce a new lower bound based on partial knowledge about the final conformation (namely the distribution of H-monomers to layers).

Keywords: structure prediction, protein folding, lattice models, HP-model

1. Introduction

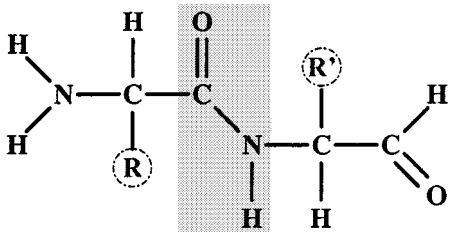
Proteins are sequences composed of an alphabet of 20 amino acids. An amino acid is a chemical group of the form



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 Cα), 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-Cα-bond (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 (which are displayed on so-called Ramachandran plots).

Using this freedom of rotation, the protein can *fold* into a specific three-dimensional structure (called conformation). In 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. The protein structure prediction problem is the problem of determining the native structure of a protein, given its sequence of amino acids. On the one hand, there are physical/chemical method for determining the native structure. One standard procedure consists of producing a pure solution containing only the protein, then crystallizing the protein, followed by an x-ray crystallography. The major limitation is the crystallization process. One needs enough material of the protein, and the solution containing the protein must be very pure (since the purity of the solution determined the exactness of the analysis). This step itself is very time consuming. The crystallization step itself is limited to a subclass of proteins (e.g., in general the interesting membrane proteins do *not* crystallize at all).

For this reason, the protein structure prediction with computer methods is one of most important problems in computational biology. There are several results showing that structure prediction problem is NP-hard. These results indicate that it is unlikely that one will find a general, efficient algorithm for solving this problem. But the situation is even worse, since one does not know the general principles why natural proteins fold into a native structure. Artifical proteins usually don't have a native structure (i.e., there is no stable structure that will be achieved by the protein).

To attack this problem, simplified models have been introduced, which became a major tool for investigating general properties of protein folding. An important class of simplified models are the so-called lattice models. The 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 positions in a lattice, and
4. a simplified energy function

There are different lattices, which can approximate real proteins arbitrarily close. The simplest used lattice is the cubic lattice, where every conformation of a lattice protein is a self-avoiding walk in $\mathbb{Z}^3$. A discussion of lattice proteins can be found in [8]. There are a lot 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 [21], [1], [10], [20], [14], [11], [12], [2], [17].

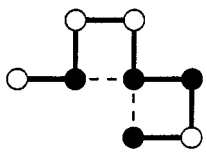
An important representative of lattice models is the HP-model, which has been introduced by [15], [16]. In this model, the 20 letter alphabet of amino acids (and the corresponding manifoldness of forces between them) is reduced to a two letter alphabet, namely H and P. H represents *hydrophobic* amino acids, whereas P represent *polar* or hydrophilic amino acids. The energy function for the HP-model is given by the matrix

	H	P
H	-1	0
P	0	0

,

which simply states that the energy contribution of a contact between two monomers is -1 if both are H-monomers, and 0 otherwise. Two monomers form a *contact* in some specific conformation if they are not connected via a bond, but occupy neighboring positions in the conformation (i.e., the distance vector between their positions in the conformation is a unit vector). A conformation with *minimal energy* (in the following called *optimal conformation*) is just a conformation with the maximal number of contacts between H-monomers. Just recently, the structure prediction problem has been shown to be NP-complete even for the HP-model [5], [7].

A sample conformation for the sequence PHPPHHPH in the two-dimensional lattice with energy -2 is



The white beads represent P-, the black ones H-monomers. The two contacts are indicated via dashed lines. A more complex conformation in $\mathbb{Z}^3$ of another HP-sequence is shown in Figure 1.

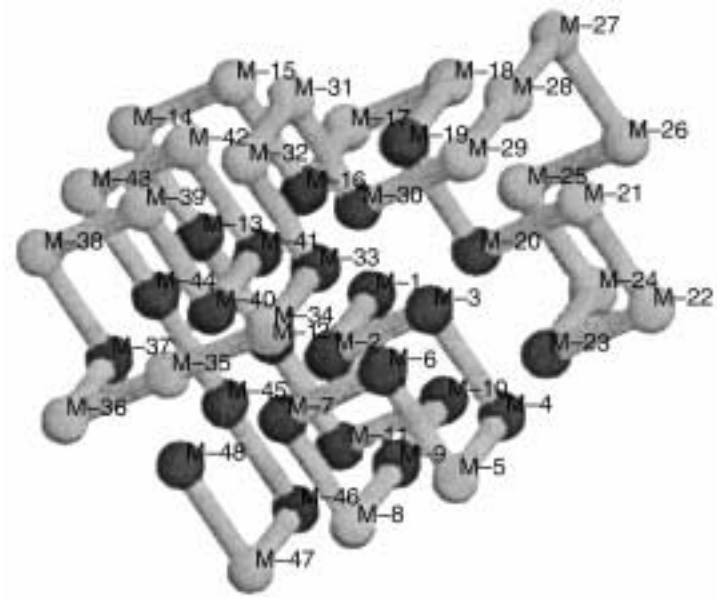


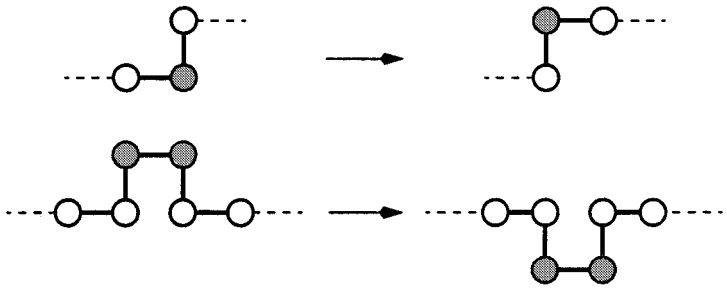
Figure 1. Native conformation of an HP-sequence of length 48 found by our algorithm. H-monomers are displayed using dark beads.

Problem 1.1 (Problem 1.1 (HP Structure Prediction Problem).) Given a sequence $s \in \{H, P\}^*$, find a conformation of s in $\mathbb{Z}^3$ having maximal number of contacts between H-monomers.

An example of the use of lattice models is is the work by Šali, Shakhnovich and Karplus [21]. They investigate under which conditions a protein folds into a native structure. For this purpose, they have performed the following computer experiment on proteins in the cubic lattice:

1. Generation of 200 random sequences of length 27.
2. Determination of the native structures on the $3 \times 3 \times 3$-cube. The $3 \times 3 \times 3$-cube has exactly 27 position, which was the reason for using a sequence length of 27 (in a later paper [10], the authors considered proteins of length 125).
3. Simulation of protein folding on the lattice model using a Monte-Carlo method with Metropolis criteria. The Monte-Carlo method is as follows. Initially, a random conformation of the sequence is generated. Starting from this initial conformation, the algorithm performs so-called Monte Carlo steps in order to search for the minimal conformation. A single Monte Carlo step consists of the following operations: First, a local move is selected at random until a move is found that produces a valid conformation (i.e., a self-avoiding conformation). Two examples of allowed

moves are



Here, the positions of the shaded monomers are changed, and the positions of the other monomers are kept unchanged.

Second, the resulting conformation is evaluated according to the Metropolis criterion. If the energy of the resulting conformation is lower than the energy of the previous one, then the conformation is always accepted. Otherwise, the conformation is accepted by random, where the probability of accepting the new conformation depends on the energy difference.[2] For every initial conformation, 50,000,000 Monte Carlo steps are performed.

Now a protein folds in that framework, if the Monte Carlo method finds its native conformation. The authors have found that a protein folds if there is a energy gap between the native structure and the energy of the next minimal structure. Note that the same lattice model has been used by several other people, such that [1], [20], [2], [12].

In performing such experiments, it is clear that the quality of the predicted principle depends on several parameters. The first is the quality of the used lattice and energy function. The second, and even more crucial point, is the ability for finding the native structure as required by Step 2. For the energy function used by [21], there is no *exact* algorithm for finding the minimal structure. This problem can be overcome by restricting the search for the native structure on the $3 \times 3 \times 3$-cube, as done in Step 2. But this approach has some drawbacks: 1.) The energy function had to be biased to hydrophobicity (i.e., the average contact energy between any two amino acids must be attractive; only then one gets proteins whose native structure is on the $3 \times 3 \times 3$-cube with high probability (see [21])); 2.) even then, it is not guaranteed that the minimal conformation is on this cube (for examples in the HP-model see [24]); 3.) the length of the proteins cannot be arbitrarily chosen.

1.1. Previous Work

In the literature, several algorithms were proposed for the HP-model, and we will discuss some of them. One class of algorithm consists of heuristic methods. There are heuristic approaches such as the hydrophobic zipper [9], the genetic algorithm by Unger and

Moult [19] and the chain growth algorithm by Bornberg-Bauer [6]. Another example is an approximation algorithm as described in Hart and Istrail [13], which produces a conformation, whose number of contacts is known to be at least $\frac{3}{8}$ of the optimal number of contacts, in linear time. And there is one exact algorithm, namely the CHCC (Constrained Hydrophobic Core Construction) of Yue and Dill [22], which finds all optimal conformations.

Our algorithm is motivated by the CHCC-algorithm [22]. The main idea of CHCC is that the surface area of the hydrophobic core is easier to estimate (given partial information about the final conformation) than the number of HH-contacts, and that the core surface area and the number of contacts are related one-to-one. Using this observation, in a first step, CHCC enumerates all possible shapes of the region containing all H-monomers of the given sequence (i.e., all core shapes). This enumeration is done in a way such that core shapes with a smaller surface area are enumerated before core shapes with a larger surface area. For every core shape, CHCC enumerates all positions of the monomers that fit into the given core shape. CHCC uses some conditions (or constraints) to reduce the size of the search tree. The major drawback of CHCC is that it is a specialized algorithm tailored for the HP-model in the cubic lattice. It is not seen how ideas from CHCC can be used for other energy functions or lattices.

1.2. Contributions of the Paper

We provide a declarative formulation of the HP-structure prediction problem in order to be able to extend the algorithm to other lattice models. Using our approach, we were able to extend the algorithm to a lattice model with more complex energy functions [4], namely the HPNX-model [6]. We are currently working to extend this formalization to another lattice, namely the face-centered-cubic lattice (FCC).

We have transformed the protein structure prediction problem to a constraint minimization problem with finite domain variables, Boolean variables, and reified constraints. We have then implemented this constraint problem using the language Oz [18].

Since a simple constraint formulization does not provide an efficient search algorithm, we investigate ways to prune the search trees. First, we introduce redundant constraints that allow to prune the search tree by removing invalid search branches (i.e., branches that do not lead to a valid conformation) early. Second, we give a search strategy that enumerates low energy conformations earlier in the search tree. This is important, since the overall search algorithm is a combination of branch-and-bound with generate-and-constraint. Thus, finding low energy conformations early in the search tree is necessary for the branch-and-bound. Another problem, which is not discussed in this paper, but in [3], is the exclusion of symmetries. The reason is that for every solution found in the search tree, there are 47 symmetric solutions that are found by the search procedure if no method for excluding symmetries is applied (for details, see [3]).

To use the branch-and-bound approach, it is necessary to find bounding functions for the energy of a conformation, given partial information derived in the search so far. For this purpose, we introduce a new lower bound on the surface of the set of all H-monomers given the distribution of H-monomers to planes orthogonal to the x-axis (i.e.,

the planes can be described by the equation $x = c$). This results in an upper bound on the number of contacts, and therefore on the energy. The lower bound on the surface uses a property of lattice models, namely that for any sequence s and any conformation of s in $\mathbb{Z}^3$, two monomers $1 \leq i, j \leq \text{length}(s)$ can form a contact if and only if $|i-j| > 1$, and i is even and j is odd, or vice versa. We have found a simple formula to calculate a lower bound without using extensive search.

1.3. Plan of the Paper

In Section 2, we introduce the basic definitions for the structure prediction problem. Furthermore, we explain that maximizing the number of contacts is the same as minimizing the surface of all H-monomers. In Section 3.1, we introduce the constraint minimization problem (since we minimize the surface to maximize the number of contacts) modeling the structure prediction problem. We introduce the variables and basic constraints, and describe the search strategy. Section 4 explains, which symmetries occur in the structure prediction problem. In the following Section 5, we prove the new lower bound on the surface given the distribution of H-monomers to layers. Finally, in Section 6, we present results for some HP-sequences taken from the literature, show search times and number of search steps.

2. Basic Definitions

A sequence is an element in $\{H, P\}^*$. With s_i we denote the ith 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 euclidic 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 $\text{Contact}_s(c)$ 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 $-\text{Contact}_s(c)$.

With $\vec{e}_x$, $\vec{e}_y$ and $\vec{e}_z$ we denote the unit vectors $(1,0,0)$, $(0,1,0)$ or $(0,0,1)$, respectively. We say that two points $\vec{p}, \vec{p}' \in \mathbb{Z}^3$ are *neighbors* if $\|\vec{p} - \vec{p}'\| = 1$. This is equivalent to the proposition that $\vec{p} = \vec{p}' \pm \vec{e}$ with $\vec{e} \in \{\vec{e}_x, \vec{e}_y, \vec{e}_z\}$. Given a conformation c, the *surface* $\mathrm{Surf}_s(c)$ is defined as the number of pairs of neighbor positions, where the first is occupied by an H-monomer, but the second not. I.e.,

$$\mathrm{Surf}_s(c) = \left| \left\{ (c(i), \vec{p}) \;\middle|\; \begin{array}{cc} & s_i = H \wedge \|\vec{p} - c(i)\| = 1 \\ \wedge & \forall j : (s_j = H \Rightarrow c(j) \neq \vec{p}) \end{array} \right\} \right|$$

Figure 2 gives an example of the surface of a sample conformation in two dimensions. Yue and Dill [22] made the observation that there is a simple linear equation relating surface and energy. This equation uses the fact that every monomer has $2 \cdot d$ neighbors in the $\mathbb{Z}^d$, each of which is in any conformation either filled with either an H-monomer, a P-monomer, or left free. Let n_H^s be the number of H-monomers in s, then we have for every conformation c that

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

where $\mathrm{HHBonds}(s)$ is the number of bonds between H-monomers (i.e., the number of H-monomers whose successor in s is also a H-monomer). $2[\mathrm{Contact}_s(c) + \mathrm{HHBonds}(s)]$ counts the number of times an H-monomer has an H-monomer at the neighbor position, and $\mathrm{Surf}_s(c)$ counts the cases where the neighbor position of an H-monomer is occupied by a P-monomer, or left free. Since $\mathrm{HHBonds}(s)$ is constant for all conformations c of s, this implies that minimizing the surface is the same as maximizing the number of contacts.

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

Given a conformation, the *frame* of the conformation is the minimal rectangular box that contains all H-monomers of the sequence. Given a vector $\vec{p}$, we denote with $(\vec{p})_x$, $(\vec{p})_y$ and $(\vec{p})_z$ the x-, y- and z-coordinate of $\vec{p}$, respectively. The *dimensions* (fr_x, fr_y, fr_z)

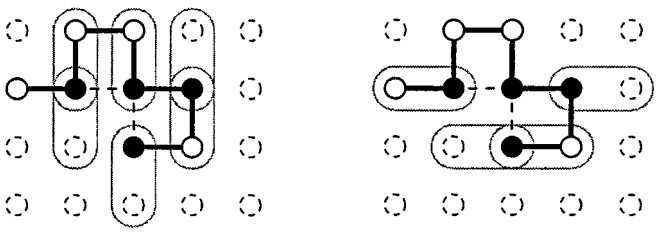


Figure 2. Vertical and horizontal contributions to the surface of a conformation in $\mathbb{Z}^2$. The surface of this conformation is 10. Every pair of positions that contribute to the surface is enclosed by a grey oval.

of the frame are the numbers of monomers that can be placed in x-, y- and z-direction within the frame. I.e.,

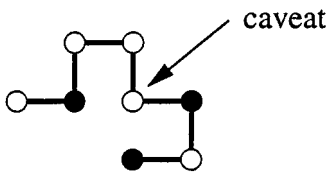
$$fr_x = \max\{|(c(i) - c(j))_{\mathrm{x}}| \mid 1 \leq i, j \leq \mathrm{length}(s) \wedge s_i = H \wedge s_j = H\} + 1$$
$$fr_y = \max\{|(c(i) - c(j))_{\mathrm{y}}| \mid 1 \leq i, j \leq \mathrm{length}(s) \wedge s_i = H \wedge s_j = H\} + 1$$
$$fr_z = \max\{|(c(i) - c(j))_{\mathrm{z}}| \mid 1 \leq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

We define s_x to be $\min\{c(i)_{\mathrm{x}} \mid 1 \leq i, j \leq \mathrm{length}(s) \wedge s_i = H\}$. s_y and s_z are defined analogously. (s_x, s_y, s_z) is called *starting point* of the frame.

A position $\vec{p} \in \mathbb{Z}^3$ is a *caveat in a conformation c of s* iff

1. $\vec{p}$ is not occupied by an H-monomer,
2. $\vec{p}$ has two neighbor positions occupied by a H-monomer that are on a straight line parallel to one of the axis.

E.g., the following conformation contains a caveat:



3. Constraint Formulation

3.1. Structure Prediction as a Constraint Problem

We start with the basic constraint formulation that underlies our search algorithm. Our algorithm is based on constraint optimization, which is the combination of two principles, namely generate-and-constraint with branch-and-bound. For using constraint optimization, we have to transform the structure prediction problem into a constraint problem. A constraint problem consists of a set of variables together with some constraints on these variables. In the following, we fix a sequence s of length n.

Our constraint problem consists of finite domain variables. We use also Boolean constraint, entailment constraints and reified constraints. With reified constraints we mean constraints of the form

$$(\mathrm{x} = 1) \leftrightarrow (\phi),$$

where x is a Boolean variable and ϕ is a finite domain constraint. x is 1 if the constraint store entails ϕ, and 0 if the constraint store disentails ϕ. A constraint store *entails* a constraint ϕ if every valuation that makes the constraint store valid also makes ϕ valid. It *disentails* ϕ if the conjunction of ϕ with the constraint store is not satisfiable. We

use also entailment constraints of the form $\phi \rightarrow \psi$, which are interpreted as follows. If a constraint store entails ϕ, then ψ is added to the constraint store. Finite domain constraints and reified constraints can be encoded directly in many modern constraint programming languages.

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 $[1..2n]$.[3] This is expressed by introducing the constraints

$$\mathtt{X}_i \in \big[1..\big(2 \cdot \mathrm{length}(s)\big)\big] \wedge \mathtt{Y}_i \in \big[1..\big(2 \cdot \mathrm{length}(s)\big)\big] \wedge \mathtt{Z}_i \in \big[1..\big(2 \cdot \mathrm{length}(s)\big)\big] \tag{3}$$

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 $i \neq j$.[4] Next we want to express that the distance between two successive monomers is 1, i.e.

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

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

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

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

But in order to use constraint optimization, we have to encode the energy function. For HP-type models, the energy function can be calculated if we know for every pair of monomers (i, j) whether i and j form a contact. For this purpose we introduce for every pair (i, j) of monomers with $i + 1 < j$ a variable $\mathtt{Contact}_{i,j}$. $\mathtt{Contact}_{i,j}$ is 1 if i and j have a contact in every conformation which is compatible with the valuations of $\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i$, and 0 otherwise. Then we can express this property in constraint programming as follows:

$$\begin{aligned} \mathtt{Xdiff}_{i,j} &= |\mathtt{X}_i - \mathtt{X}_j| \quad \mathtt{Zdiff}_{i,j} = |\mathtt{Z}_i - \mathtt{Z}_j| \\ \mathtt{Ydiff}_{i,j} &= |\mathtt{Y}_i - \mathtt{Y}_j| \quad \mathtt{Contact}_{i,j} \in \{0, 1\} \\ (\mathtt{Contact}_{i,j} = 1) &\leftrightarrow (\mathtt{Xdiff}_i + \mathtt{Ydiff}_i + \mathtt{Zdiff}_i = 1) \end{aligned} \tag{4}$$

where $\texttt{Xdiff}_{i,j}$, $\texttt{Xdiff}_{i,j}$ and $\texttt{Zdiff}_{i,j}$ are new variables. The constraint (4) is an example of a reified constraint.

Using the variables $\texttt{Contact}_{i,j}$, we can now easily encode the energy function, which is subject to constraint optimization. For the HP-model, we introduce a variable HHContacts which counts the number of contacts between H-monomers. Thus, HHContacts is defined by

$$\texttt{HHContacts} = \sum_{\substack{i+1<j\wedge \\ s(i)=H\wedge s(j)=H}} \texttt{Contact}_{i,j}. \tag{5}$$

We can now define a variable Energy, where we have the constraint

$$\texttt{Energy} = -\texttt{HHContacts}.$$

Thus, we have encoded self-avoiding walks together with a variable Energy.

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

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

$$\texttt{Energy} < E_c \tag{6}$$

is added, and the search is continued in order to find the next best conformation, which must have a smaller energy than the previous ones due to the constraint (6). This implies that the algorithm finally finds a conformation with minimal energy.

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

3.2. Pruning

Clearly, the above described constraint problem generated from a sequence s is not sufficient to yield an efficient implementation. For efficiency, one needs

1. constraints that allow the early elimination of invalid configurations, and
2. the ability for implementing a search strategy that tends to enumerate low energy conformations first.

Under 1.), we subsume both bounding functions (which allow to eliminate configuration yielding only conformations that have a higher energy than previously found conformations; such a configuration is invalid to the bound constraint (6)) and constraints that eliminate configurations which do not yield a self-avoiding walk.

3.2.1 Additional Variables and Constraints

We start with defining the additional variables used in our formulation. We have summarized the variables and their description in Table 1. As we have already mentioned, it is easier to get a bound on the H-surface instead of directly bounding the energy. Hence, we use an additional variable `Surf`, which is related to `Energy` as described in Equation (2).

With $(\texttt{Frx}, \texttt{Fry}, \texttt{Frz})$, we denote the dimension of the frame. A frame is uniquely determined by its dimensions and its starting point. Yue and Dill [22] provided a method to calculate a lower bound on the surface when all H-monomers are packed within a frame having a specific dimension. Thus, there are usually only a few frames to be searched through to find the optimal conformation, since often bigger frames have a higher lower bound for the surface than an optimal sequence found in a smaller frame. The assumption of a frame that contains all H-monomers is an efficient way of excluding many non-optimal conformations. Note that also some of the P-monomers must be included within this frame, namely those P-monomers whose left and right neighbor in the chain is an H-monomer. The reason is just that one cannot include the surrounding H-monomers into the core without also including the middle P-monomer. These P-monomers are called *P-singlets*.

Hence, we start with setting the frame dimension $(\texttt{Frx}, \texttt{Fry}, \texttt{Frz})$. If these variables are determined, we fix the frame starting point (s_x, s_y, s_z).[5] Having this, we can add for every monomer i which is either an H-monomer, or a P-singlet, the constraints

$$s_x \leq \texttt{X}_\texttt{i} \leq s_x + \texttt{Frx} - 1$$
$$s_y \leq \texttt{Y}_\texttt{i} \leq s_y + \texttt{Fry} - 1$$
$$s_z \leq \texttt{Z}_\texttt{i} \leq s_z + \texttt{Frz} - 1.$$

The remaining variables consider the different positions that a monomer can occupy. The first set of variables is related to planes parallel to the ones of the coordinate axis. An

Table 1. The finite domain variables and their description

Caveats	Boolean; is 0 if the conformation contains no caveats
Frx, Fry, Frz	dimensions of the frame
X_i, Y_i, Z_i	x-, y-, and z-coordinate of the ith monomer (where $1 \le i \le \text{length}(s)$)
E_j.seh, E_j.soh	number of even and odd H-monomers of the jth x-plane (or x-layer) in the frame, respectively (where $1 \le j \le$ Frx)
E_j.sep, E_j.sop	number of even and odd P-singlets of the jth x-layer in the frame, respectively
$\text{Elem}_i^{x,c}$	membership of monomer i in the plane defined by $x = c$, where $s_x \le c \le s_x + \text{Frx} - 1$; the constraint $\text{Elem}_i^{x,c}$ will be defined only if i is an H-monomer or a P-singlet
$\text{Htype}_{\vec{p}}$	type of the frame position $\vec{p}$; the type $\text{Htype}_{\vec{p}}$ of the position $\vec{p}$ is either 1, if it is occupied by an H-monomer, and 0 otherwise
$0_i^{\vec{p}}$	for every position $\vec{p}$ of the frame and every monomer i; $0_i^{\vec{p}}$ has Boolean value, and is 1 iff monomer i occupies the position $\vec{p}$ of the frame
$\text{Surf}_{\vec{p}}^{\vec{p}'}$	surface contribution between neighbor positions $\vec{p}$ and $\vec{p}'$; $\text{Surf}_{\vec{p}}^{\vec{p}'} = 1$ if $\vec{p}$ is occupied by an H-monomer and $\vec{p}'$ is *not* occupied by an H-monomer. Otherwise, $\text{Surf}_{\vec{p}}^{\vec{p}'} = 0$. Thus, $\vec{p}$ must be within the frame, and $\vec{p}'$ can be within the frame or outside the frame with distance 1 from the frame boundaries
Surf	complete surface of the conformation

x-layer is a plane defined by the equation $x = c$ for some integer c. *y-layers* and *z-layers* are defined analogously. For the membership of monomers to layers, we introduce additional Boolean variables. For every monomer i and every integer $s_x \le c \le s_x + \text{Frx} - 1$, we introduce a variable $\text{Elem}_i^{x,c}$. $\text{Elem}_i^{x,c}$ is 1 if the monomer is in the x-layer defined by $x = c$. Thus, we have the reified constraint

$$(\text{Elem}_i^{x,c} = 1) \leftrightarrow (X_i = c).$$

The distribution of monomers to x-layers is restricted by the following constraints valid for the cubic lattice. If two monomers i and $i+2$ are in the same x-layer, then $i+1$ must also be in the same x-layer. I.e., for every $s_x \le c \le s_x + \text{Frx} - 1$ we have

$$(\text{Elem}_i^{x,c} = 1 \wedge \text{Elem}_{i+2}^{x,c} = 1) \rightarrow (\text{Elem}_{i+1}^{x,c} = 1).$$

If two monomers i and $i+3$ are in the same x-layer, then $i+1$ and $i+2$ must also be in one x-layer. I.e., for every $s_x \le c \le s_x + \text{Frx} - 1$ we have

$$(\text{Elem}_i^{x,c} = 1 \wedge \text{Elem}_{i+3}^{x,c} = 1) \rightarrow X_{i+1} = X_{i+2}.$$

We treat y-layers and z-layers analogously.[6] Furthermore, there is a special treatment of P-singlets, which may not be buried into the core (forming a caveat) in order to achieve

an optimal conformation. Thus we can state for every P-singlet i that

$$(\mathtt{Elem}_i^{x,c} = 1 \wedge \mathtt{Elem}_{i+1}^{x,c} = 0 \wedge \mathtt{Caveats} = 0) \rightarrow \mathtt{Elem}_{i-1}^{x,c} = 1$$
$$(\mathtt{Elem}_i^{x,c} = 1 \wedge \mathtt{Elem}_{i-1}^{x,c} = 0 \wedge \mathtt{Caveats} = 0) \rightarrow \mathtt{Elem}_{i+1}^{x,c} = 1.$$

Using $\mathtt{Elem}_i^{x,c}$, we can define $\mathtt{E}_j.\mathtt{seh}$ by

$$\mathtt{E}_j.\mathtt{seh} = \sum_{i \text{ even}, s_i = H} \mathtt{Elem}_i^{x, s_x + j - 1},$$

and analogously for the others. Clearly, we have

$$\sum_{j=1}^{\mathtt{frx}} \mathtt{E}_j.\mathtt{soh} = |\{i | i \text{ odd and } s_i = H\}| \quad \sum_{j=1}^{\mathtt{Frx}} \mathtt{E}_j.\mathtt{sop} = |\{i | i \text{ odd P-singlet}\}|$$
$$\sum_{j=1}^{\mathtt{Frx}} \mathtt{E}_j.\mathtt{seh} = |\{i | i \text{ even and } s_i = H\}| \quad \sum_{j=1}^{\mathtt{Frx}} \mathtt{E}_j.\mathtt{sep} = |\{i | i \text{even P-singlet}\}|.$$

Then we have for every layer j that $\mathtt{E}_j.\mathtt{soh} + \mathtt{E}_j.\mathtt{seh} + \mathtt{E}_j.\mathtt{sop} + \mathtt{E}_j.\mathtt{sep} \leq \mathtt{Fry} \cdot \mathtt{Frz}$.

Finally, we have variables related to positions that can be occupied by monomers. Let $\vec{p} = (p_x, p_y, p_z)$ be some position and i be a monomer. The *occurrence variable* $0_i^{\vec{p}}$ is a Boolean variable that is 1 if the monomer i occupies the position $\vec{p}$, and 0 otherwise. This variable can be defined by

$$(0_i^{\vec{p}} = 1) \leftrightarrow (\mathtt{Elem}_i^{x,p_x} = 1 \wedge \mathtt{Elem}_i^{y,p_y} = 1 \wedge \mathtt{Elem}_i^{z,p_z} = 1).$$

With the constraint

$$\left(\sum_{1 \leq i \leq n} 0_i^{\vec{p}} \right) \leq 1$$

we guarantee that every position may be occupied by at most one monomer.

Since major part of the search tree is spanned over all possible assignments of monomers to positions, it is important to exclude invalid assignments as soon as possible. We do this by relating the different occurrences of neighbor positions. For every positions $\vec{p}$ and every monomer $1 < i < n$, we introduce the constraint

$$(0_i^{\vec{p}} = 1) \rightarrow \left(\left(\sum_{\vec{p}' \text{ neighb of } p} 0_{i+1}^{\vec{p}'} \right) \geq 1 \right) \tag{7}$$

and

$$(0_i^{\vec{p}} = 1) \rightarrow \left(\left(\sum_{\vec{p}' \text{ neighb of } p} 0_{i-1}^{\vec{p}'} \right) \geq 1 \right). \tag{8}$$

For $i = 1$ we introduce only the first constraint, for $i = n$ only the second. This constraint just states that i can only occupy the position $\vec{p}$ if both monomers $i-1$ and $i+1$ occupy

a neighbor position of $\vec{p}$. This generalizes the concept of the *tether length* as introduced in [22] and extended in [23], which only states which H-monomers can occupy which positions in the H-frame, not taking into account where the neighbor monomers can be placed. Thus, our constraint prunes the search tree given partial distribution of monomers to positions, which is not true for the tether constraint.

The next set of constraints relates occurrence variables and the energy variable in various ways. For every position $\vec{p}$ in the frame, we introduce the Boolean variable $\texttt{Htype}_{\vec{p}}$. This variable is 1 if the position is occupied by an H-monomer. Thus, $\texttt{Htype}_{\vec{p}}$ is defined by

$$(\texttt{Htype}_{\vec{p}} = 1) \leftrightarrow \left(\left(\sum_{1 \le i \le n \wedge s_i = H} 0_i^{\vec{p}} \right) \ge 1 \right) \tag{9}$$

Additionally, we have $\sum_{\vec{p}} \texttt{Htype}_{\vec{p}} = n_H(s)$, where $n_H(s)$ is the number of H-monomers in s.

Now we have constraints relating the type variables of positions and H-surface contributions. As already mentioned, the number of HH-contacts can be more easily approximated from the surface of all H-monomers. Thus, we introduce the Boolean variables $\texttt{Surf}_{\vec{p}}^{\vec{p'}}$ for all neighbor positions $\vec{p}$ and $\vec{p'}$, which is defined by

$$(\texttt{Surf}_{\vec{p}}^{\vec{p'}} = 1) \leftrightarrow (\texttt{Htype}_{\vec{p}} = 1 \wedge \texttt{Htype}_{\vec{p'}} = 0).$$

Of course, we get

$$\texttt{Surf} = \sum_{\vec{p}, \vec{p'} \text{ neighbours}} \texttt{Surf}_{\vec{p}}^{\vec{p'}}.$$

The variable Surf is then used to constrain the variable Energy as described by Equation (2).

Now the surface contributions and the Caveats variable can be related using reified constraints. For every line li in $\mathbb{Z}^3$ parallel to one of the coordinate axis, which intersects with the frame, we define the Boolean variable $\texttt{Caveat}_{li}$ by

$$\left(\texttt{Caveat}_{li} = 1\right) \leftrightarrow \left(\left(\sum_{\vec{p} \ne \vec{p'} \text{ on } li} \texttt{Surf}_{\vec{p}}^{\vec{p'}} \right) >: 2 \right).$$

Clearly, in the above summation we need only to consider all $\vec{p} \ne \vec{p'}$ on li which are an element of the frame or within distance 1 of the frame. Then

$$\left(\texttt{Caveats} = 1\right) \leftrightarrow \left(\sum_{\text{lines } li} \texttt{Caveat}_{li} \geqslant 1 \right).$$

3.2.2 *Search Strategy*

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

$$\texttt{Caveats} < \begin{matrix}\texttt{Frx}\\ \texttt{Fry}\\ \texttt{Frz}\end{matrix} < \begin{matrix}\texttt{E}_\texttt{j}\texttt{.seh}\\ \texttt{E}_\texttt{j}\texttt{.soh}\\ \texttt{E}_\texttt{j}\texttt{.sep}\\ \texttt{E}_\texttt{j}\texttt{.sop}\end{matrix} < \texttt{Elem}_i^{x,c} < 0_i^{\vec{p}} < \begin{matrix}\texttt{X}_i\\ \texttt{Y}_i\\ \texttt{Z}_i\end{matrix}$$

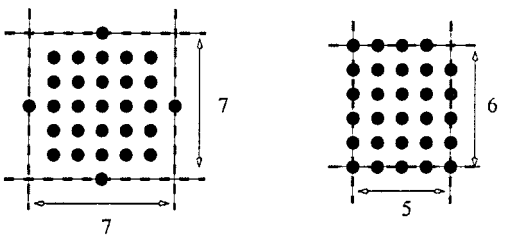
It is a good strategy to set Caveats to 0 in the first branch, since in almost every case there is an optimal conformation without a caveat. The frame dimensions are chosen ordered by surface according to the lower bound given in [22]. After having determined the variables $\texttt{E}_j\texttt{.seh}$, $\texttt{E}_j\texttt{.soh}$, $\texttt{E}_j\texttt{.sep}$, and $\texttt{E}_j\texttt{.sop}$, we calculate a lower bound on the surface. Given a conformation c, we distinguish between x-surface and yz-surface of c. The x-surface can be defined as the sum over all variables $\texttt{Surf}_{\vec{p}}^{\vec{p}'}$ where $\vec{p}-\vec{p}'$ is $\pm\vec{e}_x$. The yz-surface is

$$\texttt{Surf}_s - \text{x-surface of } c.$$

Since even H-monomers can form contacts only with odd H-monomers, a lower bound for the surface in x-direction is given by

$$\texttt{E}_1\texttt{.soh} + \texttt{E}_1\texttt{.seh} + \texttt{E}_{\texttt{Frx}}\texttt{.soh} + \texttt{E}_{\texttt{Frx}}\texttt{.seh} + \sum_{1 \le j < \texttt{Frx}} \left(|\texttt{E}_j\texttt{.soh} - \texttt{E}_{j+1}\texttt{.seh}| + |\texttt{E}_j\texttt{.seh} - \texttt{E}_{j+1}\texttt{.soh}|\right).$$

For the yz-surface, the first observation is that given a conformation c such that c contains $\texttt{E}_j\texttt{.soh} + \texttt{E}_j\texttt{.seh}$ H-monomers in the jth layer, then the surface in that layer is given by the minimal rectangle enclosing the monomers in that layer. Thus, consider the following two conformations, where the positions occupied by H-monomers in the jth layer look as follows:



Both have the property that $\texttt{E}_j\texttt{.soh} + \texttt{E}_j\texttt{.seh} = 29$. But the yz-surface in the jth layer is $2 \cdot 7 + 2 \cdot 7 = 28$ for the first conformation, and $2 \cdot 5 + 2 \cdot 6 = 22$ for the second. Hence,

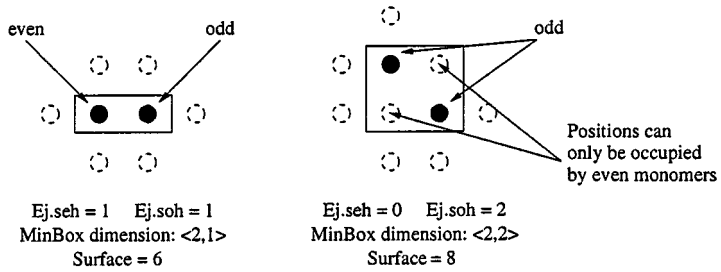


Figure 3. Minimal enclosing box for two different assignments of $\mathtt{E}_j.\mathtt{soh}$ and $\mathtt{E}_j.\mathtt{seh}$.

given $n_H^j = \mathtt{E}_j.\mathtt{soh} + \mathtt{E}_j.\mathtt{seh}$, then a lower bound for the yz-surface of the jth layer is given by $2 \cdot a + 2 \cdot b$, where

$$a = \left\lceil \sqrt{n_H^j} \right\rceil \text{ and } b = \left\lceil \frac{n_H^j}{a} \right\rceil.$$

But we can provide a better lower bound by considering the different parity of H-monomers. For any conformation c of s, every pair of odd monomers i_1 and i_2 in s satisfy $x_{i_1} + y_{i_1} + z_{i_1} \equiv x_{i_2} + y_{i_2} + z_{i_2} \mod 2$, where $(x_{i_1}, y_{i_1}, z_{i_1}) = c(i_1)$ and $(x_{i_2}, y_{i_2}, z_{i_2}) = c(i_2)$. This implies that either all odd monomers occupy odd positions (where a position $p = (x, y, z)$ is odd iff $x + y + z$ is odd), or all odd monomers occupy even positions (and similarly for the even monomers). Hence, the distribution of monomers $\mathtt{E}_j.\mathtt{seh}$ and $\mathtt{E}_j.\mathtt{soh}$ may enforce a greater yz-surface (see Figure 3 for a simple example). This will be described in Section 5.

In the following step, we assign H-monomers and P-singlets to the different layers according to $\mathtt{E}_j.\mathtt{seh}, \mathtt{E}_j.\mathtt{soh}, \mathtt{E}_j.\mathtt{sep}$ and $\mathtt{E}_j.\mathtt{sop}$. Here, we select the monomers by considering P-singlets first (since their placement yields more propagation), and then the remaining H-monomers using first-fail. If all H-monomers and P-singlets are assigned to layers, we search for the positions of these monomers within the frame. The final step consists of assigning x-, y- and z-values to all monomers which are neither H-monomers nor P-singlets.

4. Excluding Geometric Symmetries

In the structure prediction problem, we are faced with a lot of symmetries consisting of rotations, reflections and combinations of them. These symmetries are all affine mappings $S\colon \mathbb{Z}^3 \to \mathbb{Z}^3$ with $S(\vec{x}) = A_S\vec{x} + \vec{v}_S$ that map the $\mathbb{Z}^3$ onto $\mathbb{Z}^3$. I.e., the matrix A_S is an orthogonal matrix with the property that the columns $\vec{v}_1$, $\vec{v}_2$ and $\vec{v}_3$ of A_S satisfy $\forall i \in [1..3] : \vec{v}_i \in \{\pm\vec{e}_x, \pm\vec{e}_y, \pm\vec{e}_z\}$. For instance, the 90°-rotation around the z-axis is defined by the matrix

$$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Since the dimension of A_S must be 3, we have $6 \times 4 \times 2$ matrices, and henceforth 47 non-trivial symmetries. Of course, these kind of symmetries occur in any approach to structure prediction.

We have introduced a new method for excluding symmetries in [3], which is based on the constraint programming paradigm. We have successfully applied this method in our case. To our knowledge, in approaches to structure prediction that use complete enumeration, either no symmetry exclusion was done (thus, these approaches enumerate also symmetric solutions), or the search was performed by an approach that is by and large a brute force technique (brute-force enumerations are currently able only to find conformations for sequences of length smaller than 20). In the latter case, one can perform a trivial symmetry exclusion.

5. A New Lower Bound

Let c be some conformation of s with frame dimension ($\mathtt{Frx}, \mathtt{Fry}, \mathtt{Frz}$). We now distinguish between surface contribution in x-direction (x-surface), and surface contributions in the single x-layers (yz-surface). For this purpose, we define

$$\mathtt{Surf}^s_x(c) = \left| \left\{ (c(i), \vec{p}) \;\middle|\; \begin{array}{ll} & s_i = H \wedge \vec{p} - c(i) = \pm \vec{e}_x \\ \wedge & \mathrm{Htype}_c(\vec{p}) = 0 \end{array} \right\} \right|$$

$$\mathtt{Surf}^s_{(x=k)}(c) = \left| \left\{ (c(i), \vec{p}) \;\middle|\; \begin{array}{ll} & s_i = H \wedge ||\vec{p} - c(i)|| = 1 \\ \wedge & c(i)_x = k = \vec{p}_x \wedge \mathrm{Htype}_c(\vec{p}) = 0 \end{array} \right\} \right|$$

where $\mathrm{Htype}_c(\vec{p})$ is defined by $\mathrm{Htype}_c(\vec{p}) = 1 \Leftrightarrow \exists i : (s_i = H \wedge c(i) = \vec{p})$. Clearly, we have

$$\mathrm{Surf}_s(c) = \mathrm{Surf}^s_x(c) + \sum_{1 \le k \le 2 \cdot n} \mathrm{Surf}^s_{(x=k)}(c).$$

Given a point $(x, y, z) \in \mathbb{Z}^3$, we say that (x, y, z) is *odd* (resp. *even*) if $x + y + z$ is odd (resp. even). We write $(x, y, z) \equiv (x', y', z')$ iff $x + y + z \equiv x' + y' + z' \bmod 2$.

Proposition 5.1 *Let c be a conformation of s. Then $c(i) \equiv c(j)$ iff $i \equiv j \bmod 2$.*

From this we get the following lower bound on $\mathrm{Surf}^s_x(c)$, provided that we know how many even and odd monomers are placed on the jth layer. It is easy to see that the $\mathtt{E_1.soh}$ monomers generate $\mathtt{E_1.soh}$ surface points in $-$x direction. Furthermore, there are $\mathtt{E_1.soh}$ points in $+$x direction, which are candidates for surface points. But all these candidates are even points. If $\mathtt{E_1.soh} > \mathtt{E_2.seh}$, then we have minimal $\mathtt{E_1.soh} - \mathtt{E_2.seh}$ surface points in the second layer. If $\mathtt{E_1.soh} \le \mathtt{E_2seh}$, then a similar argumentation shows that we have at least $\mathtt{E_2.seh} - \mathtt{E_1.soh}$ surface points in the first layer. Continuing this, we get the following lemma.

Lemma 5.2 *Let* $(\mathtt{E}_1.\mathtt{soh}, \mathtt{E}_1.\mathtt{seh}, \ldots, \mathtt{E}_{\mathtt{Frx}}.\mathtt{soh}, \mathtt{E}_{\mathtt{Frx}}.\mathtt{seh})$ *be the number of H-monomers of c that are placed in the different layers through the frame. Then*

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

For calculating the yz-surface of a specific layer, we introduce the concept of a coloring. A coloring just states which points are occupied by some H-monomer. A *coloring* is a function $f\colon \mathbb{Z}^2 \to \{0, 1\}$. We say that a point (x, y) is colored black by f iff $f(x, y) = 1$. In the following, we consider only colorings different from the empty coloring f_e (which satisfies $\forall \vec{p} : f_e(\vec{p}) = 0$). Given a coloring f, define

$$e(f) = |\{(x, y) | f(x, y) = 1 \text{ and } x + y \text{ even}\}|$$
$$o(f) = |\{(x, y) | f(x, y) = 1 \text{ and } x + y \text{ odd}\}|.$$

The *surface* $\mathrm{Surf}(f)$ of a coloring f is defined by

$$\mathrm{Surf}(f) = \left|\left\{(\vec{p}, \vec{p}^{\,1}) | \vec{p}, \vec{p}^{\,1} \in \mathbb{Z}^2 \wedge f(\vec{p}) = 0 \wedge f(\vec{p}^{\,1}) = \wedge\right\}\right|$$

Given a pair (e, o) of integers, we define

$$\mathrm{Surf}(e, o) = \min\{\mathrm{Surf}(f) | f \text{ coloring with } e(f) = e \wedge o(f) = o\}$$

The next lemma relates the surface of colorings with the yz-surface of a conformation.

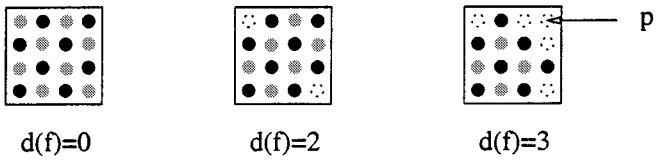
Proposition 5.3 *Let c be a conformation of s having* $\mathtt{E}_j.\mathtt{seh}$ *even and* $\mathtt{E}_j.\mathtt{soh}$ *odd points in the jth x-layer. Furthermore, let* s_x *be the minimal coordinate of the frame. Then* $\mathrm{Surf}^s_{(x=s_x+j-1)}(c) \geq \mathrm{Surf}(\mathtt{E}_j.\mathtt{seh}, \mathtt{E}_j.\mathtt{soh})$.

Thus, Lemma 5.2 together with Proposition 5.3 provide a lower bound on the surface. Since $\mathrm{Surf}(e, o) = \mathrm{Surf}(o, e)$, it is sufficient to treat the case where $e \leq o$. In the following theorem, we handle the simple case where $|e - o| \leq 1$.

Theorem 5.4 *Let* (e, o) *be a pair of integers with* $|e - o| \leq 1$. *Let* $a = \lceil \sqrt{e + o} \rceil$ *and* $b = \lceil \frac{e+o}{a} \rceil$. *Then* $\mathrm{Surf}(e, o) = 2a + 2b$.

Because $\mathrm{Surf}(e, o) = \mathrm{Surf}(o, e)$, the remaining case is to calculate $\mathrm{Surf}(e, o)$ where $e < o - 1$, without the need to search through all possible colorings f. A point $(x, y) \in \mathbb{Z}^2$ is a *caveat in* f if $f(x, y) = 0$ and (x, y) is contained in the hull (over $\mathbb{Z}^2$) of the points colored black in f. We handle only caveat-free colorings in this paper. The case of a coloring with caveats can be reduced to the caveat-free case.

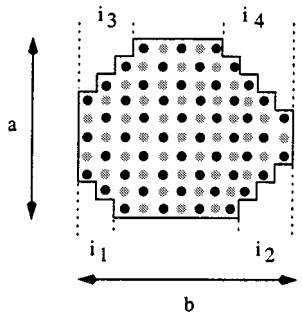
Given a coloring f, we denote with the *frame* (a, b) the maximal dimension of the coloring in y- and x-direction. Since we are considering caveat-free colorings, we get that the surface of f is $2 \cdot a + 2 \cdot b$, where (a, b) is the frame of f. Hence, we can calculate the surface of (e, o) by finding a minimal frame (a, b) such that there is a coloring of (e, o) having this frame. The first condition is clearly that $ab \geq e + o$. This condition is exactly the case that is treated in Theorem 5.4. But in the case that we have $e < o - 1$, this condition is not sufficient. The reason is that given a fixed frame (a, b), it may well be that we can color $e + 1$ even and o odd points in the frame (a, b), but not e even and o odd. E.g., consider a fixed frame of size $(4, 4)$. Grey points indicate even points, black ones odd points. We define $d(f) = o(f) - e(f)$. Then three maximal colorings for different values of $d(f)$ are



If we have the same number of even and odd points ($d(f) = 0$), then we can color at most 16 points in that frame. If $d(f) = 2$, then we can color at most 14 points. But if $d(f) = 3$, we can color at most 11 points, because we have to remove one odd position (e.g. p) before we can reduce the number of even positions. This leads to the following definition.

Definition 5.5. The partial order $\preceq$ on caveat-free colorings is defined by $f \preceq f'$ if and only if $\text{height}(f) = \text{height}(f')$, $\text{length}(f) = \text{length}(f')$, $d(f) = d(f')$ and f' extends f, i.e. $e(f) + o(f) \leq e(f') + o(f')$.

Now we have $f \preceq f'$ implies that f, f' have the same surface. The nice thing is that $\preceq$-maximal colorings have a simple normal form, from which $d(f)$ can easily be read off. An example of such $\preceq$-maximal coloring (called simple coloring) f is



(10)

Again, we use black beads for odd positions colored by f, and grey for even. (a, b) is the frame of f, and $i_1, \ldots, i_4$ are the side length of triangles excluded at the corners. The tuple $(a, b, i_1, i_2, i_3, i_4)$ is called the characteristics of this coloring (here it is

(10, 12, 2, 3, 3, 4)). We investigate simple colorings in more depth, finally showing how they can be used to provide the desired bound.

Let f be some coloring. With $\min_x(f)$ we denote the integer

$$\min\{x \in \mathbb{Z} | \exists y \in \mathbb{Z} : f(x, y) = 1\}.$$

$\max_x(f)$, $\min_y(f)$ and $\max_y(f)$ are defined analogously. Furthermore, we define

$$\text{length}(f) = \max_x(f) - \min_x(f) + 1$$
$$\text{height}(f) = \max_y(f) - \min_y(f) + 1.$$

The pair $(\text{height}(f), \text{length}(f))$ is called the *frame* of f. We say that a point $(x, y) \in \mathbb{Z}^2$ is *within the frame of* f if $\min_x(f) \leq x \leq \max_x(f)$ and $\min_y(f) \leq y \leq \max_y(f)$. Given $1 \leq i \leq \text{height}(f)$, then the ith *row* (denoted $(\text{row}(i, f))$) is the coloring r defined by

$$r(x, y) = \begin{cases} f(x, y) & \text{if } y = \min_y(f) + i - 1, \\ 0 & \text{else.} \end{cases}$$

Furthermore, we define

$$\text{indent}_l(i, f) = \min_x(\text{row}(i, f)) - \min_x(f)$$
$$\text{indent}_r(i, f) = \max_x(f) - \max_x(\text{row}(i, f)).$$

For a row $r = \text{row}(i, f)$ with $1 \leq i \leq \text{height}(f)$, we write $\text{yval}(r)$ for $\min_y(r)$ $(= \max_y(r))$. The line $y = \text{yval}(r)$ contains all points colored black by the row r. The *leftmost* (resp. *rightmost*) point in a row r is the leftmost (resp. rightmost) point colored black by r, i.e., the point $(\min_x(r), \text{yval}(r))$ (resp. $(\max_x(r), \text{yval}(r))$).

Proposition 5.6 *Let f, f' be two caveat-free colorings with $f \preceq f'$. Then* $\text{Surf}(f) = \text{Surf}(f')$.

Proof:: Since both f and f' are caveat-free, we know that $\text{Surf}(f) = 2\text{height}(f) + 2\text{length}(f)$. Similarly, we get $\text{Surf}(f') = 2\text{height}(f) + 2\text{length}(f')$. Since $f \preceq f'$, we have $\text{height}(f) = \text{height}(f')$ and $\text{length}(f) = \text{length}(f')$, which implies $\text{Surf}(f) = \text{Surf}(f')$. ■

We will show that every f can be extended to a $\preceq$-maximal coloring f' (which has the same surface by the last proposition). This implies that the surface of $\preceq$-maximal colorings extending f is a lower bound on the surface of f. To calculate the surface of $\preceq$-maximal colorings, we can show, that every $\preceq$-maximal coloring f has a simple form, as e.g. shown in (10). The properties of this simple form is defined as follows.

Definition 5.7. Let f be a caveat-free coloring with $d(f) > 1$. Then f is called *simple* iff it satisfies the following conditions:

1. for all $1 \leq i < \text{height}(f)$ we have

$$\text{indent}_l(i,f) \neq 0 \vee \text{indent}_l(i+1,f) \neq 0 \Rightarrow |\text{indent}_l(i+1,f) - \text{indent}_l(i,f)| = 1 \quad (11)$$

$$\text{indent}_r(i,f) \neq 0 \vee \text{indent}_r(i+1,f) \neq 0 \Rightarrow |\text{indent}_r(i+1,f) - \text{indent}_r(i,f)| = 1. \quad (12)$$

2. the leftmost and the rightmost point of the first and the last row are odd.

Proposition 5.8 *Let f be a simple coloring. Then*

- $\text{indent}_l(i,f) \neq 0 \Rightarrow$ *the leftmost point of row*(i,f) *is odd;*
- $\text{indent}_r(i,f) \neq 0 \Rightarrow$ *the rightmost point of row*(i,f) *is odd;*

Proof:: Follows directly from Condition 2 together with Condition 1 of the definition of a simple coloring. ■

Definition 5.9. Let f be a simple coloring with frame $(a,b) = (\text{height}(f), \text{length}(f))$. Then the tuple

$$(a, b, \text{indent}_l(1,f), \text{indent}_r(1,f), \text{indent}_l(a,f), \text{indent}_r(a,f))$$

is called the characteristics of f. A tuple $(a, b, i_1, i_2, i_3, i_4)$ is called a characteristics if it is the characteristics of some simple coloring.

First, we show some easy correlation between simple colorings and their characteristics.

Proposition 5.10 *A simple coloring f is uniquely determined (up to translation) by its characteristics, i.e., for two simple colorings f, f' having the same characteristics, there is a vector $\vec{v} \in \mathbb{Z}^2$ such that*

$$\forall \vec{p} \in \mathbb{Z}^2 : \big[(f(\vec{p}) = 1) \Leftrightarrow \big(f'(\vec{p} + \vec{v}) = 1\big)\big].$$

Proof (sketch):: E.g., consider the left lower corner. Now Equation (11) implies that the rows 1 to $i+1$ have left indents $i, i-1, \ldots, 0$, where $i = \text{indent}_l(1,f)$. The same holds for the other corners. Since f simple, this uniquely determines f (up to translation). ■

Proposition 5.11 *Let $C = (a, b, i_1, i_2, i_3, i_4)$ be a tuple. Then C is a characteristics if and only if*

1. $a - i_1 - i_3 \geq 1$, $a - i_2 - i_4 \geq 1$, $b - i_1 - i_2 \geq 1$ *and* $b - i_3 - i_4 \geq 1$;

2. *and*

$$\begin{aligned}
&a \text{ odd} \Rightarrow (i_1 \equiv i_3 \mod 2) \wedge (i_2 \equiv i_4 \mod 2)\\
&a \text{ even} \Rightarrow (i_1 \not\equiv i_3 \mod 2) \wedge (i_2 \not\equiv i_4 \mod 2)\\
&b \text{ odd} \Rightarrow (i_1 \equiv i_2 \mod 2) \wedge (i_3 \equiv i_4 \mod 2)\\
&b \text{ even} \Rightarrow (i_1 \not\equiv i_2 \mod 2) \wedge (i_3 \not\equiv i_4 \mod 2)
\end{aligned}$$

Proof (sketch):: Claim 1 follows directly from the definition of a characteristics of a simple coloring. Claim 2 follows from the fact that the leftmost and rightmost point of the first and last row must be odd, which implies that the first and last row must have an odd number of points colored black. The same argument can be applied to the first and last column. ■

Corollary 5.12 *Let $(a, b, i_1, i_2, i_3, i_4)$ be a characteristics. Then the top and bottom point of the first and last column are odd.*

The advantage of a simple coloring f is that one can easily calculate $e(f)+o(f)$ and $d(f)$ out of the characteristics, as stated in the following theorem.

Theorem 5.13 *Every coloring f can be extended to a simple coloring f' with the same surface such that $d(f) = d(f')$. Let f be a simple coloring with characteristics $(a, b, i_1, i_2, i_3, i_4)$. Then*

$$\begin{aligned}
\text{Surf}(f) &= 2a + 2b\\
e(f) + o(f) &= ab - \sum_{j=1}^{4} \frac{i_j(i_j+1)}{2}\\
d(f) &= \frac{i_1 + i_2 + i_3 + i_4}{2} + 1
\end{aligned}$$

This can be used for calculating $\text{Surf}(e, o)$ as follows. We start with the minimal frame (a, b) for $e + o$ as stated in Theorem 5.4. Then we search for numbers i_1, i_2, i_3, i_4 satisfying the above constraints. As we will show in Theorem 5.16, we do not have to search through all possible numbers for i_1, i_2, i_3, i_4. We can restrict the search to numbers i_1, i_2, i_3, i_4 where differences between the number is smaller than 2. If we find an appropriate valuation for i_1, i_2, i_3, i_4, then the $\text{Surf}(e, o)$ is given by $2a + 2b$. Otherwise, we have to search for the next bigger frame (i.e., enlarge a or b by 1). For the proof of this theorem we need an additional lemma.

Proposition 5.14 *Let f be a connected, caveat-free coloring with* $\text{height}(f) = a$. *Then*

$$\begin{aligned}
d(f) = a &- |\{i \mid \textit{the leftmost point of } \text{row}(i, f) \textit{ is even}\}|\\
&- |\{i \mid \textit{the rightmost point of } \text{row}(i, f) \textit{ is even}\}|
\end{aligned}$$

Proof:: Via induction on $a = \text{height}(f)$. For the base case $a = 1$, it holds trivially. For the induction step, let f be a coloring with $\text{height}(f) = a + 1$. Let f' be the coloring which is generated by deleting the $a+1$st row in f. Then $\text{height}(f') = a$, and we get

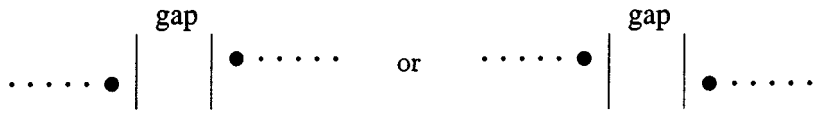
$$d(f') = a - |\{i| \text{ the leftmost point of row}(i, f) \text{ is even}\}| \\ - |\{i| \text{ the rightmost point of row}(i, f) \text{ is even}\}|$$

by induction hypothesis. Let $r = \text{row}(a+1, f)$. Then

$$d(f) = d(f') + 1 - \begin{cases} 0 & \text{if the leftmost } \textit{and} \text{ rightmost point of } r \text{ are odd} \\ 2 & \text{if the leftmost } \textit{and} \text{ rightmost point of } r \text{ are even} \\ 1 & \text{else,} \end{cases}$$

which proves the claim. ■

The remaining part is to show that a $\preceq$-maximal coloring is simple. We will first show that this holds for a subclass of caveat-free colorings, namely connected colorings. We say that a coloring f is *unconnected* if there is an i such that there is a gap between the ith and $i+1$st row, i.e., they have the form



We say that a coloring is *connected* otherwise.

Lemma 5.15 *Let f be a connected, caveat-free coloring with $d(f) > 1$ that is $\preceq$-maximal. Then f is simple.*

Proof:: First, we show that Condition 1 of Definition 5.7 is satisfied by every $\preceq$-maximal coloring. Suppose that f does not satisfy (11). Then there is some $1 \leq i < \text{height}(f)$ with

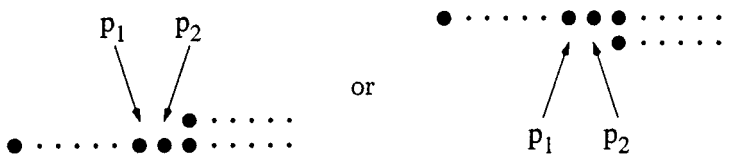
$$(\text{indent}_l(i, f) \neq 0 \vee \text{indent}_l(i+1, f) \neq 0)$$

and

$$|\text{indent}_l(i+1, f) - \text{indent}_l(i, f)| \neq 1.$$

We distinguish the following cases:

1. $|\text{indent}_l(i+1, f) - \text{indent}_l(i, f)| > 1$. Since f is connected, the ith and $i+1$st row of f have the form



with positions $p_1 = (x_1, y_1)$ and $p_2 = (x_1 + 1, y_1)$ (for some x_1, y_1) being free. Now p_1 and p_2 are positions with distance 1, which implies that they have different parities. Define f' by

$$f'(x, y) = \begin{cases} 1 & \text{if } (x, y) = p_1 \text{ or } (x, y) = p_2 \\ f(x, y) & \text{else.} \end{cases}$$

Since f is caveat-free, we know that f' is also caveat-free. Furthermore, we know that $\text{length}(f) = \text{length}(f')$ and $\text{height}(f) = \text{height}(f')$. Since p_1 and p_2 have different parity, we know that

$$e(f') = e(f) + 1 \text{ and } o(f') = o(f) + 1.$$

Hence, $d(f) = d(f')$, which implies $f \prec f'$. But this is a contradiction to the $\preceq$-maximality of f.

2. $\text{indent}_l(i + 1, f) = \text{indent}_l(i, f)$. This case can be reduced to the previous one by rotating f by 90°.

The case that f does not satisfy (12) is analogous.

Now suppose that f does not satisfy the Condition 2. Let $a = \text{height}(f)$, $b = \text{length}(f)$, $i_1 = \text{indent}_l(1, f)$, $i_2 = \text{indent}_r(1, f)$, $i_3 = \text{indent}_l(a, f)$ and $i_3 = \text{indent}_r(a, f)$. After possibly applying reflections, we can assume that the leftmost point of the first row is even. We distinguish the following cases:

1. $i_1 \neq 0$. Let $r = \text{row}(1, f)$ and define

$$p_1 = (\min_x(r) - 1, \text{yval}(r))$$

(the point left to the leftmost point of r). Then p_1 is an odd point and within the frame of f. Since $d(f) > 1$, Proposition 5.14 implies that there must be a j such that the row $r' = \text{row}(j, f)$ starts or ends with an odd point, and has non-empty indent. If row r' starts with an odd point, then take

$$p_2 = (\min_x(r') - 1, \text{yval}(r')),$$

otherwise define

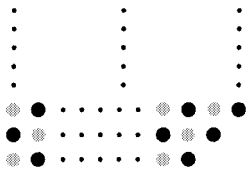
$$p_2 = (\max_x(r') + 1, \text{yval}(r')).$$

Then p_2 is an even point which is within the frame of f. Define f' by

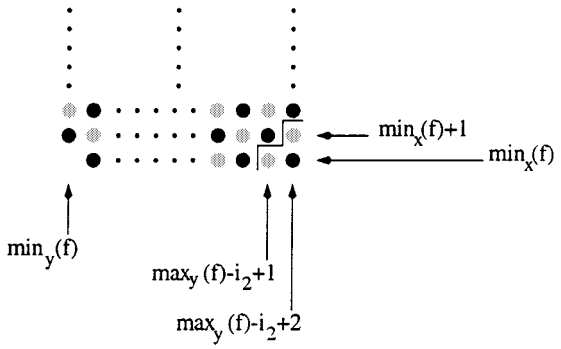
$$f'(p) = \begin{cases} f(p) & \text{if } p \neq p_1 \text{ or } p \neq p_2 \\ 1 & \text{else} \end{cases}$$

Then f' is caveat-free and connected with $f \prec f'$, which is a contradiction.

2. $i_1 = 0$. Since $d(f) > 1$, we know that by Proposition 5.14 that not all of i_2, i_3, i_4 can be lower or equal 1. Suppose that $i_2 > 2$. By the last case, we can assume that last point of the first row is odd. Hence, the first three rows of f are of the form



where again black beads indicated odd positions (x, y) with $f(x, y) = 1$, and grey beads represent even positions. Let f' be the coloring which is f except for the first three rows, where f' is of the form



I.e., f' is defined by

$$f'(x,y) = \begin{cases} 0 & \text{if } (x,y) = (\min_x(f), \min_y(f)) \\ 1 & \text{if } (x,y) = (\min_x(f), \max_y(f) - i_2 + 1) \\ 1 & \text{if } (x,y) = (\min_x(f), \max_y(f) - i_2 + 2) \\ 1 & \text{if } (x,y) = (\min_x(f) + 1, \max_y(f) - i_2 + 2) \\ f(x,y) & \text{else} \end{cases}$$

It is easy to check that $d(f) = d(f')$. Since we didn't change the height or length of f, and since we have added two points, this implies

$$f \prec f'.$$

But this is a contradiction to the $\preceq$-maximality of f. The other cases $i_3 > 2$ and $i_4 > 2$ are analogous. ■

Now we can prove our theorem.

Proof of Theorem 5.13:: The first part of the theorem follows from the definition of $\preceq$, Proposition 5.6 and Lemma 5.15.

For the second part, let f be a simple coloring as defined by the theorem. Then $e(f)+o(f)$ is just the number of points $\vec{p} \in \mathbb{Z}^2$ with $f(\vec{p}) = 1$. But this is exactly ab minus the points that are excluded at the corners. Given the indents $i_1, \ldots, i_4$, we get that we exclude exactly

$$\sum_{j=1}^{4} \frac{i_j(i_j+1)}{2}$$

points at the corners.

For proving that $d(f) = (i_1+i_2+i_3+i_4)/2+1$, we have to count the number of times the starting (resp. end point) of the row(i, f) is even (according to Proposition 5.14). This happens only if indent$_l(i, f)$ (resp. indent$_r(i, f)$) is zero. Now there are $a-i_1-i_3$ integers i with indent$_l(i, f) = 0$, and $a-i_2-i_4$ integers i with indent$_r(i, f) = 0$. Since they all have indent 0, one can see that exactly every second row starts or ends with an even point. Furthermore, Corollary 5.12 guarantees that

1. $a-i_1-i_3$ and $a-i_2-i_4$ are both odd; and
2. that there are more i's with indent$_l(i, f) = 0$ that start with an odd monomer.

The same holds for the right side. Hence, we get

$$\begin{aligned} d(f) &= a - \frac{a-i_1-i_3-1}{2} - \frac{a-i_2-i_4-1}{2} \\ &= a - \frac{a-i_1-i_3-1+a-i_2-i_4-1}{2} \\ &= a - \frac{2a-2-i_1-i_3-i_2-i_4}{2} = \frac{i_1+i_2+i_3+i_4}{2} + 1. \end{aligned}$$

■

We can even further restrict the characteristics of $\preceq$-maximal colorings.

Theorem 5.16 *Let f be a connected, $\preceq$-maximal coloring such that $d(f) > 1$. Then f has a characteristics $(a, b, i_1, i_2, i_3, i_4)$ such that*

$$\forall k, l \in [1..4] : |i_k - i_l| \leq 2.$$

Proof:: Let f be $\preceq$-maximal with characteristics $(a, b, i_1, i_2, i_3, i_4)$. Assume that f does not satisfy the condition of the lemma. I.e., there is a i_k and i_l with $k \neq l \in [1..4]$ such that

$$i_l < i_k - 2$$

After applying possibly reflection or rotation, we can assume that

$$i_1 = \min\{i_1, i_2, i_3, i_4\}.$$

and that there is an $i_k \in [2..4]$ with $i_1 < i_k - 2$. Note that by definition of characteristics, $a - i_1 - i_3 \geq 1$ and $b - i_1 - i_2 \geq 1$. We distinguish the following cases:

1. $a - i_1 - i_3 > 1$ and $b - i_1 - i_2 > 1$. By Condition 2 and Corollary 5.12, this implies that $a - i_1 - i_3 \geq 3$ and $b - i_1 - i_2 \geq 3$.

   Suppose that i_2 satisfies $i_1 < i_2 - 2$. Consider $C = (a, b, i'_1, i'_2, i_3, i_4)$ with $i'_1 = i_1 + 2$ and $i'_2 = i_2 - 2$. By Proposition 5.11 we know that C is a characteristics, which implies that there is some simple coloring f' having characteristics C. By Theorem 5.13, we know that

$$d(f') = \frac{(i_1 + 2) + (i_2 - 2) + i_3 + i_4}{2} + 1 = \frac{i_1 + i_2 + i_3 + i_4}{2} + 1 = d(f).$$

   Since f and f' have the same length and height, we need only to show that $e(f') + o(f') > e(f) + o(f)$ for showing that $f \prec f'$. This is equivalent to show that

$$nd(f', f) = e(f') + o(f') - e(f) - o(f) > 0.$$

   By Theorem 5.13, we get

$$\begin{aligned} nd(f', f) &= ab - \left(\frac{i'_1(i'_1 + 1)}{2} + \frac{i'_2(i'_2 + 1)}{2} + \frac{i_3(i_3 + 1)}{2} + \frac{i_4(i_4 + 1)}{2} \right) \\ &\quad - \left(ab - \sum_{j=1}^{4} \frac{i_j(i_j + 1)}{2} \right) \\ &= \frac{-(i_1 + 2)(i_1 + 3) - (i_2 - 2)(i_2 - 1) + i_1(i_1 + 1) + i_2(i_2 + 1)}{2} \\ &= \frac{-(i_1^2 + 5i_1 + 6) - (i_2^2 - 3i_2 + 2) + i_1^2 + i_1 + i_2^2 + i_2}{2} \\ &= \frac{-4i_1 + 4i_2 - 8}{2} \\ &= 2((i_2 - 2) - i_1) \\ &> 0 \text{ (since } i_1 < i_2 - 2 \text{ by assumption)} \end{aligned}$$

   Hence, $f \prec f'$, which is a contradiction. The other cases $i_1 < i_3 - 2$ and $i_1 < i_4 - 2$ are analogous.

2. $a - i_1 - i_3 = 1$ and $b - i_1 - i_2 > 1$. Note that by condition $a - i_1 - i_3 = 1$, we cannot just enlarge i_1 without simultaneously decreasing i_3 by the same value. Hence, we

can consider only characteristics of the forms

$$(a, b, i_1+k, i_2+l, i_3-k, i_4-l) \quad \text{or} \quad (a, b, i_1+k, i_2-l, i_3-k, i_4+l)$$

We distinguish the following cases:

(a) $i_1 < i_3 - 2$. We can then show that there is an f' with $f \prec f'$ by considering the characteristics $(a, b, i_1+2, i_2, i_3-2, i_4)$ similar to the previous case.

(b) $i_1 \geq i_3 - 2 \wedge (i_1 < i_2 - 2) \vee (i_1 < i_4 - 2)$.

Suppose that $i_1 < i_2 - 2$. Since $a - i_1 - i_3 = 1$ we get $i_3 = a - i_1 - 1$. By $i_1 = \min\{i_1 ... i_4\}$, $a - i_1 - i_3 = 1$, $i_1 \geq i_3 - 2$ and $i_1 < i_2 - 2$ we get $a - i_2 - i_4 \leq 1$. Hence,

$$i_4 \leq a - i_2 - 1 < a - (i_1 + 2) - 1 = i_3 - 2 \tag{13}$$

Consider the tuple $C = (a, b, i_1+1, i_2-1, i_3-1, i_4+1)$, which is a characteristics by Proposition 5.11. Hence, there is a simple f' having the characteristics C. We get again $d(f) = d(f')$, and we have to show that

$$nd(f', f) = e(f') + o(f') - e(f) - o(f) > 0.$$

By Theorem 5.13, we get

$$\begin{aligned} nd(f', f) &= \frac{-(i_1+1)(i_1+2) - (i_2-1)i_2 - (i_3-1)i_3 - (i_4+1)(i_4+2)}{2} \\ &\quad + \frac{i_1(i_1+1) + i_2(i_2+1) + i_3(i_3+1) + i_4(i_4+1)}{2} \\ &= \frac{-i_1^2 - 3i_1 - 2 - i_2^2 + i_2 - i_3^2 + i_3 - i_4^2 - 3i_4 - 2}{2} \\ &\quad + \frac{i_1^2 + i_1 + i_2^2 + i_2 + i_3^2 + i_3 + i_4^2 + i_4}{2} \\ &= \frac{-2i_1 + 2i_2 + 2i_3 - 2i_4 - 4}{2} \\ &= i_2 - i_1 + i_3 - i_4 - 2 \\ &> 2 + 2 - 2 = 2 \text{ since } i_1 < i_2 - 2 \text{ and } i_4 < i_3 - 2 \text{ by (13).} \end{aligned}$$

which shows that $f \prec f'$.

The case that $i_1 < i_4 - 2$ can be proved analogous to the case that $i_1 < i_2 - 2$. We can then show that $i_2 < i_3$, and prove the existence of an f' with $f \prec f'$ by using the characteristics $(a, b, i_1+1, i_2+1, i_3-1, i_4-1)$.

(c) $a - i_1 - i_3 = 1$ and $b - i_1 - i_2 = 1$. Analogous to the previous case. ■

Finally, we have to treat colorings that are caveat-free, but that are split into unconnected parts. For simplicity, we will only sketch the proofs for these kind of colorings.

Lemma 5.17 *Let f be a caveat-free coloring that is not connected. Then there is a coloring f' with $f \prec f'$.*

Proof (sketch):: Let f have frame (a, b). If one has n unconnected subparts with frames $(a_1, b_1), \ldots, (a_n, b_n)$, then the caveat-freeness of f implies that

$$a = a_1 + \cdots + a_n \quad \text{and} \quad b = b_1 + \cdots + b_n.$$

Then one can show that one finds always a characteristics for the frame (a, b) which has the same difference than the sum of differences of the characteristics of the subparts of f, but which has more points colored black. ■

6. Results

We have implemented our approach using the constraint language Oz [18]. We have tested the program on all sequences presented in [22]. For each of those we found an optimal conformation. In Table 2, we have listed the test sequences together with the optimal conformation found, its sequence length and its optimal surface. For comparison, the runtimes (on a Sun4) of the algorithm in [22] for all optimal conformations are 1 h 38 min for L1, 1 h 14 min for L2, 5 h 19 min for L3, 5 h 19 min for L4 and 20 min for L5, respectively. There is a newer, more efficient version of this algorithm reported in [23], but there are no explicit runtimes given for these or others sequences. In Table 3, we have listed the number of steps to find a first conformation (and a second, if the first was not optimal), the number of steps needed to prove optimality, and the runtime on a Pentium 180 Pro. Currently, we are working to transform our approach to a more realistic lattice model, the face-centered cubic lattice (FCC).

Table 2. Test sequences

	Sequence and Sample Conformation	*Length*	*Optimal Surface*
L1	HPPPPHHHHPPHPHPHHHPHPPHHPPH RFDBLLFRFUBULBDFLUBLDRDDFU	27	40
L2	HPPPHHHHPHPHHPPPHPHHPHPPPHP RFDLLBUURFDLLBBRURDDFDBLUB	27	38
L3	HPHHPPHHPPHHHHPPPHPPPHHHPPH RFLDLUBBUFFFDFURBUBBDFRFDL	27	38
L4	HHPHHPHHPHHHHHHPPHHHHHPPHHHHHHH RRFDBLDRFLLBUFLURFDDRFUBBUFRDD	31	52
L5	PHPPHPPHPPHPPHPPHPPHPPHPPHPPHPPHPPHP RFDBDRUFUBRBLULDLDRDRURBLDLULURBRFR	36	32^7

Below every sequence, we list an optimal conformation. Every conformation is represented as a sequence of bond directions (R = right, L = left, F = forward, B = backward, U = up and D = down).

Table 3. Search time and number of search steps for the sample sequences

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

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

The first table are the results for an earlier implementation without symmetry exclusion. The second table shows the results for the current implementation containing symmetry exclusion, which contains also some other optimisations. The main reduction in the number of search steps (and the corresponding reduction in time) is due to the symmetry exclusion. Only for L4, the older implementation achieves a better result. The reason is that for this sequence, both implementations actually do not have to perform a search to find the optimal conformation. In this special case, the symmetry exclusion is clearly an overhead, which slows down the computation.

Acknowledgment

Many thanks to Prof. Peter Clote, who got me interested in bioinformatics and inspired this research. I would like to thank Prof. Martin Karplus for helpful discussions on the topic of lattice models, and for motivating me to apply constraint programming techniques to lattice protein folding. I would like to thank Dr. Erich Bornberg-Bauer, who initiated this research, too. I would like to thank him also for explaining the biological background to me, and for many discussion and hints. Furthermore, I would like to thank Sebastian Will, who helped me to implement the constraint problem. I would also like to thank Ralph Matthes for reading a draft version of this paper, and for many helpful comments. Remaining errors are due to me, of course.

Notes

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 (which is in fact an imino acid).
2. To be precise, the new conformation is accepted with a probability $e^{(\Delta E/k_b T)}$. k_b is the Boltzmann constant, and T is the folding temperature.

3. 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.
4. This cannot be directly encoded in most constraint programming languages, but we reduce these constraints to difference constraints on integers.
5. This can always be done in a way which is compatible with $(\mathtt{Frx}, \mathtt{Fry}, \mathtt{Frz})$ and the constraint (3).
6. Mainly, the variables $\mathtt{Elem}_i^{y,c}$ and $\mathtt{Elem}_i^{z,c}$ are introduced to have a uniform description.
7. Yue and Dill [22] have stated that the optimal surface is 16, but this is a typo since the conformation they have shown for this sequence has a surface of 32.

References

1. Abkevich, V. I., Gutin, A. M., & Shakhnovich, E. I. (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., Gutin, A. M., & Shakhnovich, E. I. (1997). Computer simulations of prebiotic evolution. In *Proc. of the Pacific Symposium on Biocomputing '97*, pages 27–38.
3. Backofen, R., & Will, S. (1999). Excluding symmetries in constraint-based search. In J. Jaffar, ed., *Proceedings of 5th International Conference on Principle and Practice of Constraint Programming (CP'99)*, volume 1713 of *Lecture Notes in Computer Science*, pages 73–87. Berlin: Springer-Verlag.
4. Backofen, R., Will, S., & Bornberg-Bauer, E. (1999). Application of constraint programming techniques for structure prediction of lattice proteins with extended alphabets. *J. Bioinformatics*, 15(3): 234–242.
5. Berger, B., & Leighton, T. (1998). Protein folding in the hydrophobic-hydrophilic (HP) modell is NP-complete. In *Proceedings of the Second Annual International Conferences on Compututational Molecular Biology (RECOMB98)*, New York, pages 30–39.
6. Bornberg-Bauer, E. (1997). Chain growth algorithms for HP-type lattice proteins. In *Proceedings of the First Annual International Conferences on Compututational Molecular Biology (RECOMB97)*, Santa Fe, New Mexico, pages 47–55. New York: ACM Press.
7. Crescenzi, P., Goldman, D., Papadimitriou, C., Piccolboni, A., & Yannakakis, M. (1998). On the complexity of protein folding. In *Proceedings of STOC*, pages 597–603. Short version in *Proceedings of RECOMB'98*, pages 61–62.
8. Dill, K. A., Bromberg, S., Yue, K., Fiebig, K. M., Yee, D. M., Thomas, P. D., & Chan, H. S. (1995). Principles of protein folding—a perspective of simple exact models. *Protein Science*, 4: 561–602.
9. Dill, K. A., Fiebig, K. M., & Chan, H. S. (1993). Cooperativity in protein-folding kinetics. *Proceedings of the National Academy of Sciences USA,* 90: 1942–1946.
10. Dinner, A. R., Šali, A., & Karplus, M. (1996). The folding mechanism of larger model proteins: role of native structure. *Proceedings of the National Academy of Sciences USA,* 93: 8356–8361.
11. Govindarajan, S., & Goldstein, R. A. (1997). Evolution of model proteins on a foldability landscape. *Proteins*, 29(4): 461–466.
12. Govindarajan, S., & Goldstein, R. A. (1997). The foldability landscape of model proteins. *Biopolymers*, 42(4): 427–438.
13. Hart, W. E., & Istrail, S. C. (1996). Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal. *Journal of Computational Biology*, 3(1): 53–96.
14. Hinds, D. A., & Levitt, L. (1996). From structure to sequence and back again. *Journal of Molecular Biology*, 258: 201–209.
15. Lau, K. F., & Dill, K. A. (1989). A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22: 3986–3997.

16. Lau, K. F., & Dill, K. A. (1990). Theory for protein mutability and biogenesis. *Proceedings of the National Academy of Sciences USA*, 87: 638–642.

17. Ortiz, A. R., Kolinski, A., & Skolnick, J. (1998). Combined multiple sequence reduced protein model approach to predict the tertiary structure of small proteins. In *Proc. of the Pacific Symposium on Biocomputing '98*, volume 3, pages 375–386.

18. Smolka, G. (1995). The Oz programming model. In J. van Leeuwen, ed., *Computer Science Today*, volume 1000 of *Lecture Notes in Computer Science*, pages 324–343. Berlin: Springer-Verlag.

19. Unger, R., & Moult, J. (1993). Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, 231: 75–81.

20. Unger, R., & Moult, J. (1996). Local interactions dominate folding in a simple protein model. *Journal of Molecular Biology*, 259: 988–994.

21. Šali, A., Shakhnovich, E., & Karplus, M. (1994). Kinetics of protein folding. *Journal of Molecular Biology*, 235: 1614–1636.

22. Yue, K., & Dill, K. A. (1993). Sequence-structure relationships in proteins and copolymers. *Physical Review E*, 48(3): 2267–2278.

23. Yue, K., & Dill, K. A. (1995). Forces of tertiary structural organization in globular proteins. *Proceedings of the National Academy of Sciences USA*, 92: 146–150.

24. Yue, K., Fiebig, M., Thomas, P. D., Chan, H. S., Shakhnovich, E. I., & Dill, K. A. (1995). A test of lattice protein folding algorithms. *Proceedings of the National Academy of Sciences USA*, 92: 325–329.