On the Complexity of String Folding *

Mike Paterson
Department of Computer Science
University of Warwick
Coventry CV4 7AL, England

Mike.Paterson@dcs.warwick.ac.uk

Teresa Przytycka
Department of Mathematics and Computer Science
Odense University
DK 5230 Odense M, Denmark

przytyck@imada.ou.dk

Abstract

A *fold* of a finite string S over a given alphabet is an embedding of S in some fixed infinite grid, such as the square or cubic mesh. The *score* of a fold is the number of pairs of matching string symbols which are embedded at adjacent grid vertices. Folds of strings in two- and three-dimensional meshes are considered, and the corresponding problems of optimizing the score or achieving a given target score are shown to be NP-hard.

1 Introduction

The motivation for the string-folding problems considered here lies in computational biology. Prediction of the three-dimensional structure of a protein from its known linear sequence of amino acids is an important practical open problem, which seems to be extremely challenging. The way in which a protein folds determines many of its biological and chemical properties. A natural approach is to look for a spatial configuration achieving a minimum free energy level. The energy is determined by such factors as the number of chemical bonds established between amino acid residues in the sequence and the number of hydrophobic interactions.

While most people would expect that finding a minimum energy configuration would be computationally intractable, previous results to this effect are very limited. Ngo and Marks [6] consider the problem of embedding a string of atoms of length exponential in the input size. The string is described by giving its length (in binary) and the locations and descriptions of the small number of special atoms along the chain which do *not* have the default geometric characteristics. The energy which is to be minimised is based on the dihedral angles of the (non-default) bonds. The proof uses a transformation from the PARTITION problem (see [3]). Unger and Moult [7] use a model like ours in that they embed a string over an arbitrary alphabet into the three-dimensional mesh, but their "distance function" is very artificial. In effect it forces the active subsequence of a string to lie in one straight line. It is then easy to design a transformation from the OPTIMAL LINEAR ARRANGEMENT problem (see [3]). Fraenkel's construction [2] is much more elaborate. He uses a model with charged atoms (his alphabet is $\{-1, 0, 1\}$), where the interactions are taken between all pairs of atoms embedded at adjacent vertices of the (two- or three-dimensional) mesh. He uses a reduction from 3-DIMENSIONAL MATCHING (see [3]). The most significant limitation of this result, in comparison with our problem, is that the object to be embedded is a (rather exotic) graph rather than just a string.

*Supported in part by the ESPRIT Basic Research Action Programme of the EC under contract No. 7141 (project ALCOM II). Most of this work was done while the second author was visiting the University of Warwick.

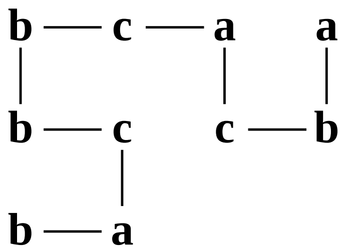

Figure 1: Fold of **bacbbcacba** in $\mathbb{Z}^2$ with a score of 4

We too prove an NP-completeness result for a much-simplified model, in which we attempt to capture rather more of the essential character of the protein-folding problem. The protein molecule is represented by a string of symbols, a bond can be made only between a pair of identical symbols, and we seek an embedding of the given string in a grid so as to maximise the number of pairs of matching symbols at adjacent grid points. This version of the folding problem involves a mixture of combinatorial, geometric and topological considerations. Hart and Istrail [4] have given an approximation algorithm for the "hydrophobic-hydrophilic" model over the same grids. This corresponds to our model, but with a binary alphabet in which the only matches counted are between adjacent 1's.

2 Preliminaries

A fixed infinite grid $\mathbb{G}$ is given. In our paper this graph will be either the two-dimensional square mesh $\mathbb{Z}^2$ or the three-dimensional cubic mesh $\mathbb{Z}^3$, though other grids such as the triangular mesh or tetrahedral meshes would also be of interest. A given finite string S, of length n say, is to be embedded in $\mathbb{G}$. A *fold of S in $\mathbb{G}$* is an injective mapping from $[1, \ldots, n]$ to $\mathbb{G}$ such that adjacent integers map to adjacent nodes of $\mathbb{G}$. (Each node of $\mathbb{Z}^2$ has four neighbours.) The *score* of a fold of S in $\mathbb{G}$ is the number of bonds in the fold, where a *bond* is a pair of identical symbols mapped to adjacent nodes of $\mathbb{G}$. For convenience we do not count a pair of successive identical symbols in S as forming a bond. In Figure 1 we show a fold of $S =$ **bacbbcacba** in $\mathbb{Z}^2$ which has a score of 4. Readers may like to verify that this score is maximal and that the fold achieving it is unique up to the obvious symmetries. The maximum score for a fold of S in $\mathbb{Z}^3$ is 5 however.

We define the following recognition version of the problem of finding an optimal fold.

STRING-FOLD
Instance: A finite string S, an integer k, and a grid $\mathbb{G}$.
Question: Is there is a fold of S in $\mathbb{G}$ with a score of at least k?

Note that the alphabet of symbols is not fixed but is implicitly part of the instance. We have been unable to extend our results to deal with a fixed alphabet.

A variety of different models for bonding are possible. We may wish to represent "neutral" elements which can form no bonds. However, any symbol which occurs exactly once in S obviously has this property and can be regarded as a neutral or *blank* symbol. For notational clarity and convenience we will use a single new symbol, $*$, for such blanks. Also we may want bonds to be formed between pairs of "complementary" symbols, $\mathbf{a}_+$ and $\mathbf{a}_-$ say. This feature comes automatically, though somewhat artificially, for string folding in bipartite grids such as $\mathbb{Z}^d$, since alternate symbols in the string must map into grid nodes of opposite parity. Since adjacent grid nodes have opposite parity, we can regard a symbol $\mathbf{a}$ as being an $\mathbf{a}_+$ or an $\mathbf{a}_-$ according to whether it occurs in an even or odd position in S.

Our main results are that STRING-FOLD is NP-complete when $\mathbb{G}$ is either $\mathbb{Z}^2$ or $\mathbb{Z}^3$. These will be proved in Sections 3 and 4. The proofs involve transformations from two known NP-complete problems, 3SAT [1, 3] and "planar" 3-satisfiability, P3SAT [5, 3]. These useful technical problems are defined as follows.

3SAT
Instance: A set $X = \{x_1, \ldots, x_m\}$ of variables and a collection $B = \{C_1, \ldots, C_k\}$ of clauses over X such that each clause has three literals.
Question: Is there a truth assignment for X such that each clause in B has at least one true literal?

The planarity condition for P3SAT is given in terms of the following associated graph. Given clauses B and variables X as above, the graph $G(B) = (V, E)$ is given by $V = B \cup X$, and $E = E_1 \cup E_2$ where:
$E_1 = \{(C_i, x_j) | x_j \in C_i \text{ or } \neg x_j \in C_i\}$ are the so-called *variable-clause* edges and
$E_2 = \{(x_j, x_{j+1}) | 1 \leq j \leq m\} \cup \{(x_m, x_1)\}$ are the so-called *variable-variable* edges.
P3SAT
Instance: A set X of variables and a collection B of clauses as in 3SAT, such that $G(B)$ is planar.
Question: Is B satisfiable?

3 STRING-FOLD in $\mathbb{Z}^2$

In this section, we show that the STRING-FOLD problem for the grid $\mathbb{Z}^2$ is NP-complete, by using a transformation from P3SAT.

Given a planar formula B with k clauses, we construct a string $S = s_1 s_2 \ldots s_n$ such that S can be embedded with score $f + k$ if and only if B is satisfiable, where f is a value that follows from the construction of the string. The string S is composed from several substrings designed separately. In combining substrings, a symbol $*$ from one string will sometimes be replaced by a symbol from another string so that the corresponding substrings fit together. A substring composed entirely of $*$'s is called a *flexible substring.*

Conflict graph. Given a set $\mathcal{S}$ of substrings of S, a *conflict graph* for $\mathcal{S}$ is a graph G, whose vertex set is the set of pairs (i, j), $1 \leq i < j \leq n$, such that $s_i = s_j$ and i, j have opposite parity. Thus the vertices of G represent potential bonds. The edge set of G has the property that if $((i, j), (i', j'))$ is an edge of G then, for every fold of S, the bonds (i, j) and (i', j') are mutually exclusive.

Note that we do not require that a conflict graph be maximal, i.e., that every pair of conflicting bonds is represented by an edge of G. Any conflict graph can be used to give an upper bound for the maximum score of a fold of $\mathcal{S}$, since the following property follows immediately from the definition.

Lemma 1 *If G is a conflict graph for a set $\mathcal{S}$ of substrings of S then in any fold of S the bonds of $\mathcal{S}$ form an independent set in G.*

Our basic tool for constructing conflict graphs is given by the following easy lemma.

Lemma 2 *Let $S = U a_1 V b_1 W a_2 X b_2 Y$, where U, V, W, X, Y are substrings, and $a_1 = a_2$, $b_1 = b_2$ are pairs of symbols of opposite parity. If $(|W| > (|V| + |X|)^2$ and $\min\{|U|, |Y|\} > (|V| + |W| + |X|)^2$ then, in any fold of S, the bonds (a_1, a_2) and (b_1, b_2) exclude each other.*

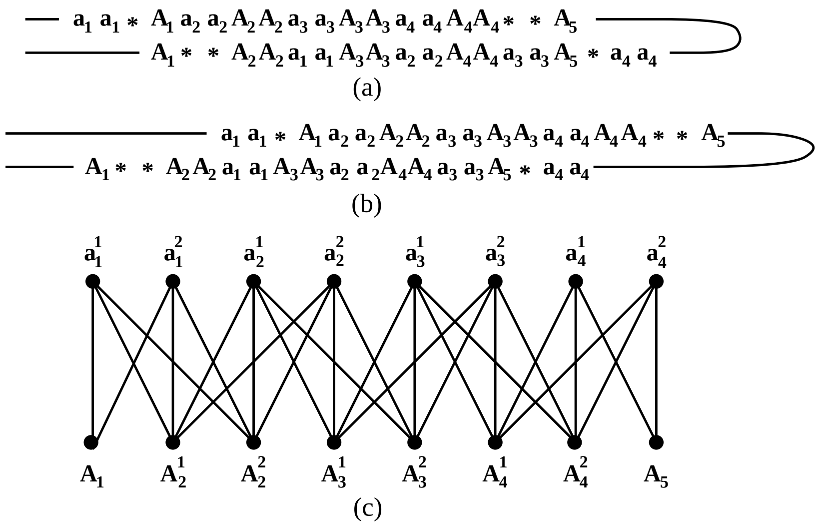

Figure 2: (a),(b) Two optimal folds of a shift-line, and (c) the conflict graph (for $m = 4$)

We will apply Lemma 2 in cases where V and X have lengths bounded by a small constant, and W can easily be made large enough to satisfy the first inequality. Since we can extend U and Y suitably just by adding extra blank symbols at the beginning and end of the string S, the inequalities in the statement of the lemma can be easily satisfied.

Shift-lines. A *shift-line of order* m is a pair of strings, $S_1 = a_1a_1 * A_1a_2a_2A_2A_2 \dots a_ia_iA_iA_i \dots a_ma_mA_mA_m * *A_{m+1}$, $S_2 = A_1 * *A_2A_2a_1a_1A_3A_3a_2a_2 \dots A_iA_ia_{i-1}a_{i-1} \dots A_mA_ma_{m-1}a_{m-1}A_{m+1} * a_ma_m$, such that S_1 and $\overline{S_2}$ are substrings of S with S_1 preceding $\overline{S_2}$, where $\overline{S}$ denotes the reversal of S and the occurrences of the first symbols of S_1 and S_2 have the same parity.

Lemma 3 *The maximum score of a shift-line of order m is $2m$. Furthermore the maximum score is achieved only by the fold whose bonds are formed by the pairs of upper case letters or by the fold using the pairs of lower case letters.*

Proof: It is easy to confirm that the score $2m$ is obtained by a fold that has one of the two types of bonds specified in the statement of the lemma (see Figure 2(a) and (b)).

To show that $2m$ is the maximum possible score and that it is attained by no other set of matches, we use the conflict graph argument. Denote by s^i the i^{th} occurrence of a symbol s in the string S_1 or in the string S_2, and construct a conflict graph G_m for $\{S_1, S_2\}$ as follows (see Figure 2(c)). G_m has $4m$ vertices $\{A_1, A_2^1, \dots, A_m^1, A_m^2, A_{m+1}, a_1^1, a_1^2, \dots a_m^1, a_m^2\}$ where vertex A_i^t (resp. a_i^t), $t = 1, 2$, corresponds to the bond $\{A_i^t, A_i^t\}$ (resp. $\{a_i^t, a_i^t\}$). For any i, $1 < i \le m$, the graph induced by $A_i^1, A_i^2, a_{i-1}^1, a_{i-1}^2, a_i^1, a_i^2$ is a complete bipartite graph with the bipartition corresponding to upper case and lower case letters. Furthermore A_1 is adjacent to a_1^1 and a_1^2 and A_{m+1} to a_m^1 and a_m^2. By Lemma 2, the resulting graph G_m is a conflict graph.

We claim that G_m has precisely two maximum size independent sets: $I_1 = \{a_1^1, a_1^2, \dots a_m^1, a_m^2\}$ and $I_2 = \{A_1, A_2^1, A_2^2, \dots A_m^1, A_m^2, A_{m+1}\}$. To show this, let I be an arbitrary independent set. Regarding Figure 2(c) as a $2 \times 2m$ array, no two elements of I can be in the same column, and so $|I| \le 2m$. Furthermore, if $|I| = 2m$ then every column of the array contains exactly one element of I. If I has some element from each row of the array, then it must have a pair of such elements lying in adjacent columns. However this is impossible since any such pair is adjacent in G_m. This contradiction concludes the proof. $\square$

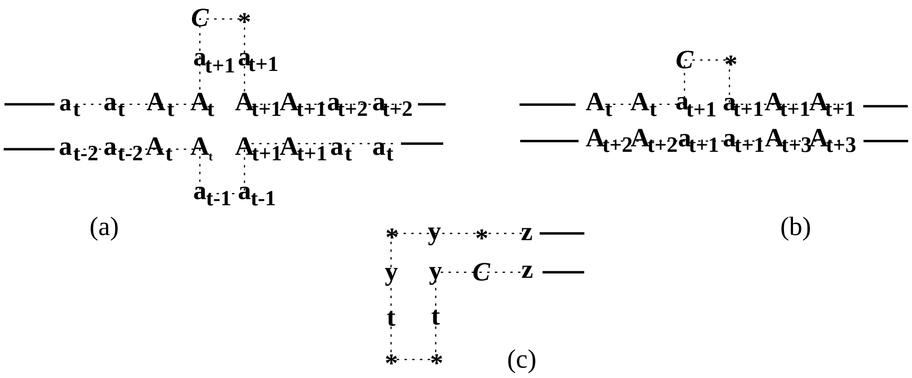

Figure 3: (a),(b) Two optimal folds of a shift-line containing an exposer that correspond to assignments to the variable which satisfy and do not satisfy clause C_i respectively. For a negated variable the pair $C_i *$ would be inserted between symbols denoted by *upper case* letters. (c) Representation of a variable trap for clause C_i.

The fact that there are only two possible sets of bonds for an optimal fold of a shift-line does not imply that there are only two embeddings of a shift-line. Each optimal embedding of a shift-line can be visualised as a double chain of matched intervals of length two interleaved with double flexible strings of length two. This flexibility would allow a shift-line which starts horizontally to move up or down in a series of steps, while maintaining its horizontal orientation. We also observe that it was not essential for the shift-line to be constructed from two continuous substrings. Between any pair of adjacent matching symbols in a shift-line we can insert a short subsequence, built over a set of symbols disjoint from the symbols in the shift-line. Provided the inequalities for Lemma 2 are satisfied, Lemma 2 still holds. Similarly, a longer subsequence could be inserted provided that there are several bonds due to be formed between matching pairs of symbols in the initial and final parts of the subsequence. Then, if the ends of the subsequence are not embedded closely enough together to apply Lemma 2, enough of these bonds would be lost to negate any possible advantage gained by a non-standard embedding of the shift-line.

The reduction. We are ready to prove the main theorem of this section. Most structures will be presented by example and picture rather than formally.

Theorem 1 *STRING-FOLD is NP-complete for the grid* $\mathbb{Z}^2$.

Proof: We will have a unique symbol C_i corresponding to each clause C_i, and we design the string S so that there is an occurrence of C_i, each with the same parity, corresponding to each variable in the i^{th} clause, and there is one occurrence of C_i with the opposite parity corresponding to the clause itself. The occurrences are designed so that, in any optimal fold, C_i can form at most one bond. In an optimal fold, the creation of a bond corresponds to a satisfying literal for the corresponding clause.

Each variable is represented by a shift-line. For each occurrence of the variable in a clause, say C_i, we insert in the shift-line a so-called *exposer* (see Figure 3(a),(b)) containing the symbol C_i. Each clause C_i is represented by a substring called a *trap* containing the symbol C_i with opposite parity to the parity of the C_i's in the exposers (see Figure 3(c)). The two optimal folds of the shift-line allow or prevent an extra bond between the C_i from the exposer and the C_i from the trap.

It remains to show how the substrings corresponding to variables and clauses are composed to

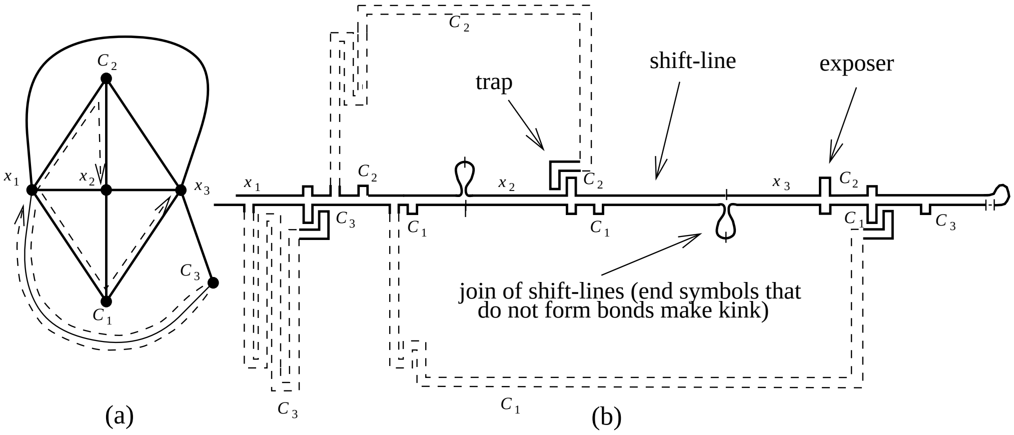

Figure 4: (a) The graph $G(B)$ for formula $B = C_1 \wedge C_2 \wedge C_3$, where $C_1 = x_1 \vee \neg x_2 \vee \neg x_3$, $C_2 = x_1 \vee x_2 \vee \neg x_3$ and $C_3 = \neg x_1 \vee x_3$, and (b) its representation on $\mathbb{Z}^2$. The embedding of flexible strings realizing the connection between the base of a trap and the exposer corresponds to non-intersecting paths in $G(B)$ denoted in diagram (a) with dashed lines.

form the string S. First we construct the string $T = [S_1^i]_{i=1}^n * [\overline{S_2^{n+1-i}}]_{i=1}^n$ where S_1^i, S_2^i is a shift-line representing x_i. We think of $[S_1^i]_{i=1}^n$ as the *top side* of T and of $[\overline{S_2}^{n+1-i}]_{i=1}^n$ as the *bottom side* of T. The shift-lines are constructed using disjoint sets of symbols.

Next, we add exposers and traps to T. Consider a fixed planar embedding of the graph $G(B)$. See Figure 4. The traps and exposers that correspond to the clauses in the interior of the cycle $(x_1, \ldots, x_n)$ in the embedding of $G(B)$ will be added to one side (say the top side of T), and traps and exposers corresponding to the clauses embedded in the exterior of this cycle on the opposite side. The order of placing the exposers on each side of the shift-line for x_i is defined by the cyclic ordering of edges adjacent to x_i. On the top side of the shift-line the exposers follow the cyclic order of the variable–clause edges about x_i inside the variable cycle, from the edge $(x_i, x_{(i-1) \bmod n})$ to $(x_i, x_{(i+1) \bmod n})$, and similarly for the bottom side and the edges outside the variable cycle.

A clause C_i is represented by a trap containing symbol C_i. The trap is attached using flexible strings to any of the shift-lines corresponding to a variable that occurs in C_i. We call the place of attachment of a trap the *base* of the trap. The trap is attached to the corresponding shift-line next to the exposer with the symbol C_i (i.e., the substring separating the base of the trap and the exposer cannot contain other traps or exposers — it can contain only shift-line symbols).

We assume that all shift-lines will be embedded horizontally along the first coordinate axis. The lengths of the flexible substrings from bases to traps are sufficient for each trap to reach any exposer corresponding to occurrences of variables in its clause, independently of other traps reaching their exposers.

By the planarity of $G(B)$, there are no topological obstructions to such simultaneous connection between pairs of trap-bases and corresponding exposers (see Figure 4(a)). However we have to take into account that the flexible strings use some area and have fixed length. This presents no significant problem and details are omitted from this abstract.

Let f be the number of bonds in an optimal fold of the string S' obtained from S by replacing all symbols C_i $(1 \leq i \leq k)$ with $*$. The bonds of S' are called *construction bonds*. A fold of S can have $f + k$ bonds only if it is possible to create a bond between each symbol C_i from its trap and some symbol C_i from an exposer. It is impossible to create a bond with a non-exposed exposer without breaking some construction bond. Similarly, a symbol C_i from a trap cannot be adjacent to two

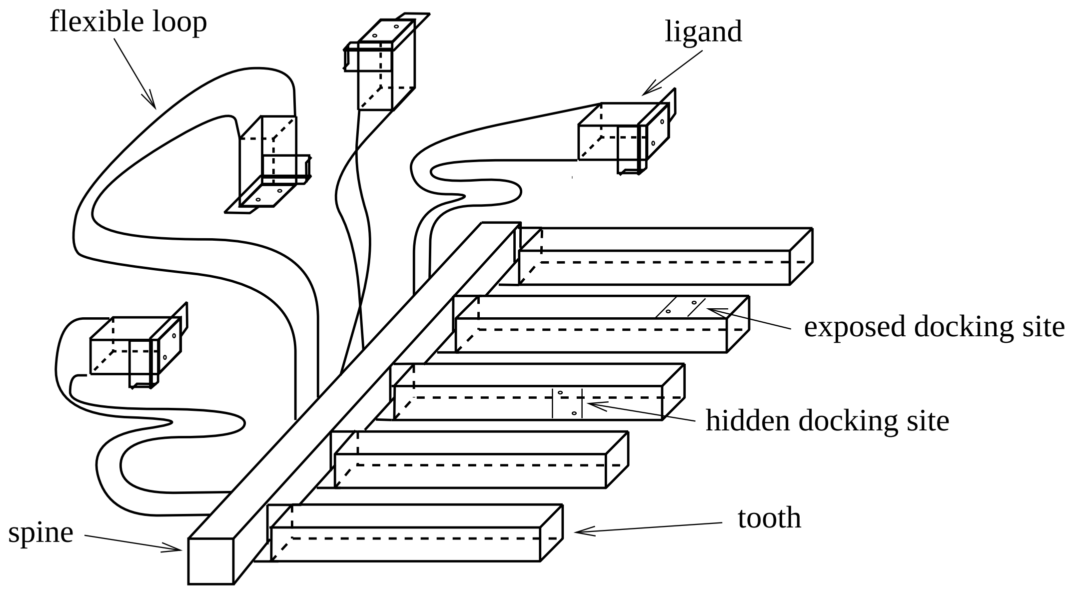

Figure 5: Overall construction

symbols C_i from two exposers without breaking at least two construction bonds in the exposers or the trap. Thus B is satisfiable if and only if an optimal fold of S has $f + k$ bonds. $\square$

4 STRING-FOLD in $\mathbb{Z}^3$

In three dimensions we will use a different mechanism to transmit the truth value of a variable to its clauses. In a faint echo of the biological origins of the problem, we represent each variable by a double helix. The truth value is expressed by the chirality (left- or right-handedness) of the helix. The proof of NP-completeness uses a reduction from 3SAT.

The intended layout of the string S is as a doubled string, following the overall shape of a "comb", in which each "tooth" consists of a helix corresponding to a variable. The teeth are attached to the spine of the comb in such a way that, although they are constrained to lie parallel to each other in a regular planar array, there is an independent choice of chirality for each tooth. There are docking points on each tooth corresponding to each clause in which a literal of that variable occurs. Depending on the chirality of the embedding chosen for that tooth, a docking site is either exposed on the top or bottom surface of the comb and available for docking, or hidden in the crevice formed with an adjacent tooth.

For each clause, there is, attached to the comb at any suitable place, a long flexible loop at the end of which is a "ligand" corresponding to that clause. The loop is long enough for the ligand to dock with any one exposed docking site corresponding to that clause on a tooth. The target score is such that it can be attained if and only if there is a choice of chirality for each tooth (i.e., truth value for each variable) such that at least one docking site for each clause is exposed (i.e., each clause is satisfied by at least one literal). An impression of the overall structure is shown in Figure 5.

We proceed to describe the components in more detail. A string $S = S_1\overline{S_2}$ is called a (*rooted*) *helix of order* m if $S_1 = (s^1_0, s^1_1, s^1_2, \ldots s^1_{6m+2}) = (1, [4i, 4i-1, *, 4i+1, 4i+2, 4i+1]^m_{i=1}, *, 4m+3)$, and $S_2 = (s^2_0, s^2_1, \ldots s^2_{6m+1}) = (2, 3, [4i, 4i+1, 4i, 4i+3, 4i+2, *]^m_{i=1})$. (See the bold lines in Figure 6(a).)

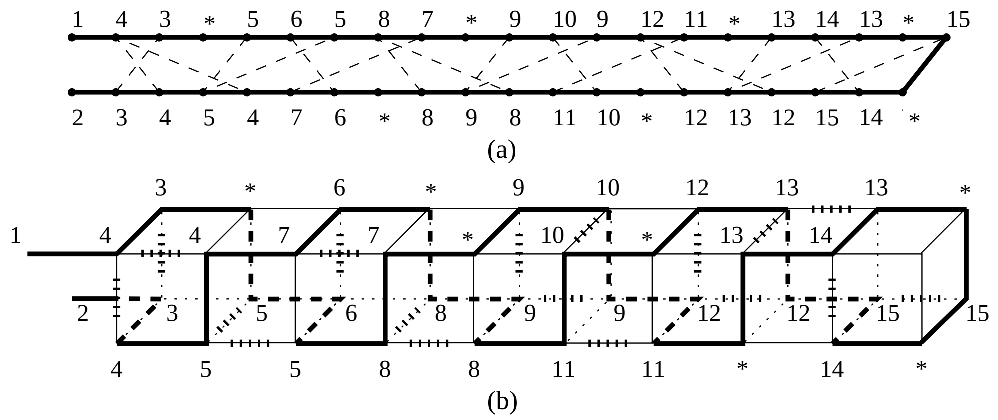

Figure 6: The double helix of order $m = 3$

Note that the parity of s_0^1 is the same as the parity of s_0^2.

Lemma 4 *The maximum score of an embedding of a helix of order m is $6m$. Furthermore, given a fixed embedding of the first and last edges of the helix such that (s_0^1, s_1^1) and (s_0^2, s_1^2) lie parallel along opposite edges of a unit cube, there are exactly two folds that achieve the maximum score. These consist of a right-handed and a left-handed double helix.*

Proof: The score of an embedding of S is maximized if all pairs of identical symbols of different parity are adjacent. Such an embedding is possible (see Figure 6(b)) and we will argue that there are only two different optimal embeddings. The second optimal embedding is similar to the one presented in Figure 6(b) but the two strings twist around each other in the opposite sense.

Let the *bond graph* $B(S) = (V, E)$ be the graph with set of vertices V equal to the set of elements of S, and such that $(x, y) \in E$ if and only if x, y are either two consecutive elements of S, or x and y are equal symbols with different parity, which implies that one is from S_1 and the other from S_2. (See Figure 6(a).) Then any optimal fold of S corresponds to a grid embedding of $B(S)$ such that each edge is embedded on an edge of the grid. In the full paper we complete the proof of Lemma 4 with an inductive argument. $\square$

Now we are ready to describe the details of the overall construction. All elements of the "comb" are built up using helices. We think of each helix as a sequence of adjacent cubes. The face defined by the last two elements of S_1 and the last element of S_2 (in Figure 6(b) the rightmost face) is called the *extremal face*. The outside faces along the length of the helix are called *external faces*. By Lemma 4, depending on the chirality of the optimal embedding each external face is in one of two pairwise perpendicular positions. Using this observation, the "comb" is designed as follows:

teeth: Each variable x corresponds to a tooth of the comb and is represented by a helix. For each occurrence of the variable in a clause C_i, a pair of "diagonal" $*$ symbols (i.e., two $*$'s that are endpoints of an external face diagonal) in the helix are replaced by the pair of symbols (C_i, D_i). This is done in such a way that in any optimal fold all the pairs of symbols corresponding to non-negated occurrences of x are embedded on external faces perpendicular to those for negated occurrences of x.

spine: The spine of the comb is built from one helix. For attaching teeth, we use the observation that any optimal embedding of a helix contains external faces of the form $A = (a, *, b, \hat{*})$ such that $a, *, b$ are consecutive in the string and there are exactly two a symbols in the

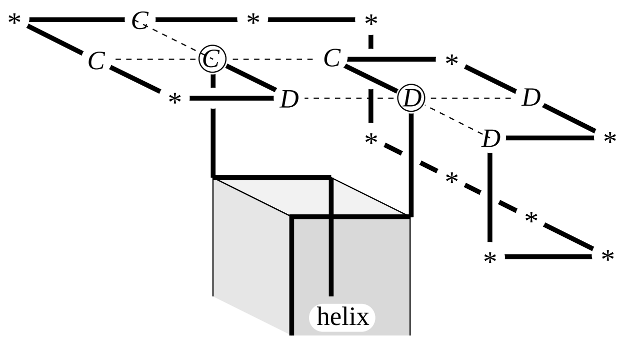

Figure 7: Fold of special trap attached to a helix

whole string, these having opposite parity (for example, see face $(11, *, 13, *)$ in Figure 6(b)). Since faces with the above properties occur periodically on a helix, the teeth can be placed regularly along the base parallel to one another. Let the new tooth be formed from the string $\hat{1}T\hat{2}$, a variable helix with initial and final symbols $\hat{1}$ and $\hat{2}$. We replace the substring $a * b$ by $a\,\hat{2}\,\hat{1}\,T\,\hat{2}\,a\,b$. The new bond (a, a) forces the new symbol a to be embedded in the same position as the $*$ symbol replaced. To make sure that the tooth is not loose, we replace the $\hat{*}$ in A by $\hat{1}$.

ligand: Referring to Figure 6(b), we see that the extremal face has a diagonal pair of $*$ symbols. We let each ligand corresponding to a clause $C = C_i$ be a helix at least as long as the distance between the teeth, to which is attached the special trap illustrated in Figure 7. (We use (C, D) for (C_i, D_i).) The subsequence $*(4m + 3)*$ which occurs on the extremal face of the helix is replaced by $\tau = C\,D * C * C * * * * * * * D * D * C\,D$. The sequence τ has four occurrences of C and four of D and, as Figure 7 shows, τ can be embedded so that all three potential (C, C) bonds and all three potential (D, D) bonds are simultaneously achieved. The effect of such an embedding is to leave the circled occurrences of C and D with only one free edge in the grid with which to make a further bond with an exposed symbol on a tooth. The target score will be set so as to require four (C, C) bonds and four (D, D) bonds. The ligand is attached to the comb using flexible strings long enough to reach any exposed pair (C, D). In this way a ligand can dock without a penalty with any (but with at most one) such pair exposed by a variable tooth.

This completes the description of the structure used to establish our second main result.

Theorem 2 *STRING-FOLD is NP-complete for the grid* $\mathbb{Z}^3$.

Proof: One aspect of the proof is much simpler than in the corresponding theorem for $\mathbb{Z}^2$. The target score which is set requires the simultaneous formation of the *maximum possible* number of bonds for every alphabet symbol. We have shown that for most parts of the structure this requirement imposes an embedding which is unique up to mirror symmetry.

Without loss of generality we can take one fixed embedding for the spine of the comb. Each tooth is attached to fixed points on the spine by a pair of edges, which are the first and last edges in these helices. By Lemma 4 there are just two embeddings of each tooth relative to the spine. The choice between these two embeddings determines which pair of opposite long faces of the tooth are exposed in the plane of the comb and which are hidden in the gaps between successive teeth.

Each ligand has several distinct embeddings, but the essential active part of each is the diagonal pair (C, D), circled in Figure 7, which is constrained to occur at the extremal face of the helix of the ligand. This helix is designed to be long enough so that the active pair cannot reach any docking site which is on a hidden surface of a tooth (see Figure 5). The flexible loops which attach the ligands to the spine of the comb offer no obstruction to achieving any docking.

The target score is reached if and only if every ligand achieves its bonds, and this is possible if and only if there is an orientation of each tooth so that every ligand has a corresponding exposed docking site to dock with. This last condition is equivalent to there being a choice of truth value for each variable in the instance of 3SAT such that each clause is satisfied by at least one of its literals. □

5 Open Problems and Conclusion

To obtain our results, we needed to allow an alphabet of unbounded size. The principal open problem that remains is to resolve the complexity of STRING-FOLD in $\mathbb{Z}^2$ and $\mathbb{Z}^3$ for the "hydrophobic-hydrophilic" model considered by Hart and Istrail [4]. This corresponds to a binary alphabet in which only one symbol forms bonds. An intermediate problem, which still seems challenging, is to extend our NP-hardness results to some fixed finite alphabet.

The grids $\mathbb{Z}^2$ and $\mathbb{Z}^3$ that we have used are bipartite, and parity arguments were helpful in maintaining control over the possible embeddings. This feature is not in keeping with the biological motivation and more realistic models. It would be a significant advance to extend our results to triangular and tetrahedral grids, which do not have the convenience of bipartiteness.

We expect our results to be of interest more to computer scientists than biologists since our model is very restricted and omits so many of the important characteristics of the protein-folding problem. The grid we impose does not capture the subtlety of molecular geometry, the model of bonds is much too simple.

It is not clear whether the biological motifs (e.g., the docking of ligands and the double helix) arose naturally in our solution or suggested themselves subconsciously because of the biological background to the problem. We hope that our examples and open questions will stimulate others to tackle string-folding problems in a more biologically realistic model.

Acknowledgements

We are grateful to Aviezri Fraenkel for introducing this type of problem to us, and to William Hart and Sorin Istrail for making available to us an early copy of their paper.

References

[1] S.A. Cook. The complexity of theorem-proving procedures. *Proc.* 3$^{\text{rd}}$ *ACM Symp. on Theory of Comp.* (1971), 151–158.

[2] A.S. Fraenkel. Complexity of protein folding. *Bull. Math. Biology* 55, (1993), 1199–1210.

[3] M. Garey and D. Johnson. Computers and Intractability. (W.H. Freeman and Co. 1979).

[4] W.E. Hart and S. Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal. *Proc.* 27$^{\text{th}}$ *ACM Symp. on Theory of Comp.* (1995), 157–168.

[5] D. Lichtenstein. Planar formulae and their uses. *SIAM J. Computing* 11, (1982), 329–343.

[6] J.T. Ngo and J. Marks. Computational complexity of a problem in molecular structure prediction. *Protein Engineering* 5, (1992), 313–321.

[7] R. Unger and J. Moult. Finding the lowest free energy conformation of a protein is an NP-hard problem: proof and implications. *Bull. Math. Biology* 55, (1993), 1183–1198.