On the Complexity of Protein Folding

Pierluigi Crescenzi, Deborah Goldman, Christos Papadimitriou
Antonio Piccolboni, Mihalis Yannakakis

Abstract

We show that the protein folding problem in the two-dimensional H-P model is NP-complete.

1 Introduction

Proteins are polymer chains consisting of monomers of twenty different kinds. Much of the genetic information in the DNA contains the sequence information of proteins, with three nucleotides encoding one monomer. In turn, proteins in an organism *fold* to form a very specific geometric pattern, known as the protein's *native state*. It is this geometric pattern that determines the macroscopic properties, behavior, and function of a protein. It is in general reasonably stable and unique.

The mapping from DNA sequences to monomer sequences is simple and very well-understood. In contrast, the mapping from the sequence of a protein to the geometric configuration of its native state —the "second half of the genetic code" [8]— is much more intricate and complex, and less understood; it has been the subject of intense investigation for decades. It seems clear that the forces underlying protein folding are the interactions between their monomers; recently, the view that *non-local* interactions dominate this process has been gaining ground [4]. To test this and other hypotheses concerning protein folding, researchers resorted to *simplified models* of proteins, mathematical abstractions of proteins that hide many aspects and exaggerate the effect of others; analysis and computer simulation of such models can then be compared to experimental results with actual proteins, to determine whether the emphasized aspects are indeed the dominant ones.

Perhaps the most successful and best-studied such model, and the one with apparently the best match with experiments[1], is the *two-dimensional hydrophilic-hydrophobic model*, or H-P model, proposed by Dill [3]. In this model it is assumed that the protein is a sequence of 0s and 1s, and folding entails embedding the sequence in the two-dimensional lattice (see Figure 1). Each such folding is evaluated with a *score*, equal to the number of pairs of 1s that are adjacent in the lattice without being adjacent in the sequence; for example, in Figure 1 the score is five, corresponding to the five pairs of 1s connected by dotted lines. The score captures a simple model of energy minimization, in which the "hydrophobic" 1s tend to be close to each other and thus avoid exposure, while 0s are neutral. It is assumed in this model that the native folded state is the one that maximizes score. It is therefore an interesting problem, given a sequence of 0s and 1s, to

[1] Chan and Dill [2] state that *"for chain lengths for which exhaustive enumeration is possible (up to about 30 monomers), two-dimensional models more accurately represent the physically important surface-interior ratios of proteins than do three-dimensional models."*

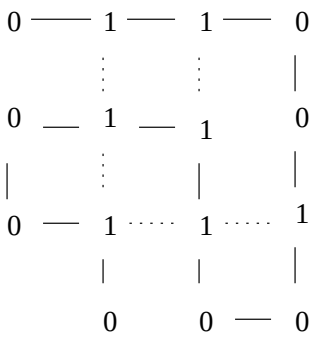


Figure 1: Embedding in the two-dimensional lattice.

find the embedding on the lattice that maximizes score. *In this paper we prove that this problem is NP-complete* (Theorem 3).

That proteins fold so as to minimize energy has been accepted for decades. This view quickly leads to a puzzling aspect of the problem, known as *Levinthal's paradox,* which can be paraphrased as follows "How can a folding protein choose so quickly among so many possible foldings the one with minimum energy?" [4]. Our result can thus be seen as a more compelling restatement of that paradox, since it implies that finding the optimum folding in the two-dimensional HP-model —the simplest abstraction of the protein folding problem one finds in the literature, and presumably a vast simplification of the true detailed 3-dimensional energy minimization problem in actual proteins— is NP-complete, that is to say, among the provably hardest problems of the sort alluded to by the paradox, in which we must optimize among an astronomical population of states.

There have been several NP-completeness results related to protein folding in the literature. A few years ago, several authors pointed out that certain general restatements of the problem, in which monomers attract or repel each other in ways that are general and can be used in encoding, are NP-complete [5, 10, 13]. More interestingly, it was proved in [11] that a combinatorial generalization of the H-P model to an infinite alphabet, of which one symbol is neutral like H-P's 0 symbol, and the score counts the number of adjacencies of elements with the same symbol, is NP-complete. More recently, [9] improved this to a finite, albeit very large alphabet. The present result is the first to settle the complexity of the simple two-dimensional H-P model actually proposed in the literature as the ultimate simplification of the protein folding problem. The H-P problem has been attacked from the point of view of approximation algorithms [6]; the present result sheds little light on this aspect of the problem, as our reduction is not in any interesting way approximation-preserving.

Our reduction is from the Hamilton cycle problem. As is common in previous proofs of weaker results, we start by showing that the folding problem for *sets of sequences* (that is, when many sequences are to be optimally folded) is NP-complete (Theorem 1 in Section 2). We then proceed to establish the result for a single sequence, by resorting to certain interesting variants of the planar Hamilton cycle problem (Theorem 3 in Section 3). In our proof we utilize an idea of Trevisan [12], whereby graphs can be embedded in the hypercube so that adjacency is captured by Hamming distance.

Our proof captures one of the basic intuitions of the H-P model, namely that hydrophobic monomers

will tend to form a large "sphere" (in the two-dimensional lattice, a large hydrophobic square). Impurities in this sphere then must be aligned optimally to maximize score, and it is the complexity of this alignment that our proof captures. Finally, in Section 4 we briefly discuss a version of our proof (in fact, without the planarity complication) which settles the NP-completeness of the three-dimensional version of the the protein folding problem in the H-P model —and in fact, the MAXSNP-completeness of the problem of minimizing losses in three dimensions. We were recently informed that, independently, Berger and Leighton [1] proved that the three-dimensional protein folding problem in the H-P model is NP-complete; in fact, the approximability implications of their result are stronger than ours.

2 The multistring folding problem

The *two-dimensional lattice* is the graph, (Z^2, L), with node set Z^2 (all points in the Euclidean plane with integer coordinates), and edges all pairs in $L = \{((x,y),(x',y')) : |x-x'|+|y-y'| = 1\}$. Consider a set of strings $S = \{s_1, \ldots, s_m\}$ from the alphabet $\{0,1\}$. A *folding* of these strings is an embedding of S into the lattice, that is to say, a one-to-one mapping f from the set $\{(i,j) : 1 \leq i \leq m, 1 \leq j \leq |s_i|\}$ to Z^2 such that for all $1 \leq i \leq m, 1 \leq j \leq |s_i| - 1$ we have $(f(i,j), f(i,j+1)) \in L$. Fix a folding f; the points $f(i,j)$ and $f(i,j+1)$ are called *f-neighbors.* An edge of the lattice $\{(x,y),(x',y')\} \in L$ is said to be a *loss* if (a) these points are not f-neighbors, and (b) exactly one of these two points is the image under f of a pair (i,j) such that the jth symbol of s_i is a 1. Each position in a string containing a one, and which is not the first or the last, can participate in zero, one, or two losses.

The MULTISTRING FOLDING PROBLEM is the following: given a set of strings $s_1, \ldots, s_m \in \{0,1\}^*$ and an integer E, is there a folding with E or fewer losses? If, as is the case in the strings we construct, no string starts or ends in a 1, then it is easy to see that the total score of a folding is equal to twice the number of 1's, minus the losses, divided by two; hence, minimizing losses is the same as maximizing score, the traditional way of stating the protein folding problem.

In this section we prove the following theorem:

Theorem 1 *The* MULTISTRING FOLDING PROBLEM *is NP-complete.*

In the next section we shall show that the problem remains NP-complete even if there is only one string.

2.1 Description of the reduction

We start from the following NP-complete problem: Given a graph $G = (V, E)$ with nodes of degree four or less, and two nodes $v_1, v_n \in V$, is there a Hamilton path from v_1 to v_n?

As a preliminary step in our reduction, we first map the nodes in G to the hypercube according to a map used by Trevisan in [12]. Using Hadamard codes, he showed that there exists a function, which we call T, mapping the n nodes of the graph to codewords in $\{0,1\}^{8n}$ such that the images

of two unconnected nodes have Hamming distance strictly greater than two nodes connected by an edge; in particular, applying T to the nodes of G we find, for $i \neq j$ in $\{1, 2, \ldots, n\}$:

1. If $\{v_i, v_j\}$ is an edge in G, then $d_H(T(v_i), T(v_j)) = \frac{7}{2}n$.

2. If $\{v_i, v_j\}$ is not an edge, then $d_H(T(v_i), T(v_j)) = 4n$.

(Here we assume that n is even.) Notice that if there is a Hamilton path from v_1 to v_n in G, then there is a Hamilton path from $T(v_1)$ to $T(v_n)$ in the Hamming space of length $\frac{7}{2}(n-1)n$; otherwise, if there is no Hamilton path from v_1 to v_n in G, then any Hamilton path from $T(v_1)$ to $T(v_n)$ must have length strictly greater than $\frac{7}{2}(n-1)n$. We note, finally, that the function T may be chosen so that $T(v_1)$ and $T(v_n)$ contain at most as many zeros as $T(v_i)$, for any $i \in \{1, \ldots, n\}$.

We now construct a set of strings S and an integer E, such that there is a Hamilton path from v_1 to v_n in G if and only if there is a folding of the strings in S with at most E losses. The allowed number of losses is

$$E = 7(n-1)n.$$

As for the set of strings S, let $L = 180n^{14}$. S will contain L strings, $s_1, \ldots, s_L$, such that the first L/n strings correspond to node v_1, the second L/n to node v_2, and so on. All strings, with the exception of s_1 and s_L, will be constructed similarly and so that strings corresponding to the same city are identical. Define $q = \lceil \sqrt{LE} \rceil$ and let c be an even positive integer to be specified later. Let d (suggesting *dense*) denote the string $d = \prod_{i=1}^{L/90} 1^8 0$, let m (suggesting *middle*) denote the string $m = \prod_{i=1}^{2E-16n} 1^{cq} 0$, and, for $i \in [n]$, let $t(i)$ (suggesting the *Trevisan code*) denote the string $t(i) = \prod_{l=1}^{8n} (1^{cq} T(v_i)_l)^2$. The description of the strings s_2 through s_{L-1} follows: for $k \in \{2, \ldots, L-1\}$ and $i = \lceil \frac{nk}{L} \rceil$,

$$s_k = 0^{L^4} d 1^{L/10} d 1^{L/5 - 2E(cq+1)} m t(i) 1^{2L/5} d 0^{L^4}.$$

Notice that each string contains a prefix and suffix of L^4 zeros. There are three dense substrings of ones and zeros in each string. Finally, toward the middle of the string, there is the substring $mt(i)$ of length $2E(c\lceil \sqrt{LE} \rceil + 1)$, which we call the *sparse substring*, which consists of the sparse string m and the sparse string $t(i)$ containing two copies of the Trevisan code (interspersed between strings of ones). The substring lying between the prefix and suffix, called the *internal substring*, has total length L.

The strings s_1 and s_L, called the *flanks*, are identical, respectively, to strings s_2 and s_{L-1} except for the fact that 4 zeros are inserted between every other pair of two adjacent ones in s_2 and s_{L-1}, beginning with the first pair of ones in each maximal substring of adjacent ones. Notice that, by placing two copies of each bit of the Trevisan code next to each other (separated by ones), we have arranged for all maximal substrings of adjacent ones to have even length. Formally, setting $d' = \prod_{i=1}^{L/90} (10^4 1)^4 0$, $m' = \prod_{i=1}^{2E-16n} (10^4 1)^{cq/2} 0$, and, for $i \in [n]$,

$$t(i)' = \prod_{l=1}^{8n} \begin{cases} ((10^4 1)^{cq/2} T(v_i)_l)^2 & \text{if } T(v_i)_l = 0 \\ (10^4 1)^{cq/2} T(v_i)_l 0^4 1 (10^4 1)^{cq/2 - 1} 10^4 T(v_i)_l & \text{if } T(v_i)_l = 1 \end{cases},$$

for each $k \in \{1, L\}$ and $i = \lceil \frac{nk}{L} \rceil$,

$$s_k = 0^{L^4} d'(10^4 1)^{L/20} d'(10^4 1)^{L/10 - E(cq+1)} m' t(i)' (10^4 1)^{L/5} d' 0^{L^4}.$$

This completes the description of the reduction.

2.2 The intended folding

In this subsection we show that if there is a Hamilton path from v_1 to v_n in G, then there is a folding with at most E losses. This is rather easy; the hard direction is the opposite, sketched in the next subsection (and proved in the appendix).

Let $(v_{i_1}, v_{i_2}, \ldots, v_{i_n})$ be the order of the nodes contained in a Hamilton path, where $v_{i_1} = v_1$ and $v_{i_n} = v_n$. We construct a folding, called the *intended folding,* by arranging the non-flank strings, $s_2, \ldots, s_{L-1}$, vertically to form a $(2L^4+L) \times (L-2)$ rectangle as follows: all the strings corresponding to node v_{i_1} are placed adjacent with their first bits in the same horizontal line at the left side of the rectangle, then the strings corresponding to node v_{i_2} are placed next, and so on until the strings corresponding to node v_{i_n} complete the rectangle. The flank strings s_1 and s_L, which differ only from s_2 and s_{L-1} through the addition of groups of 4 zeros, also have vertical orientation except for the fact that the 4-zero groups are bent to the left and right, respectively. In this way, the flanks s_1 and s_L can be placed, respectively, adjacent to the left and right sides of the rectangle so that their first bits are in line with the first bits of the other strings and so that, since the 4-zero groups have been excluded, the resulting patterns of bits adjacent to the rectangle are exactly the strings corresponding to nodes v_1 and v_n. A schematic drawing of the intended folding appears in Figure 2.

Note that in the intended folding the central $L \times L$ square is composed primarily of ones with some horizontal lines of zeros and horizontal lines containing code bits running through it. It should be clear that the only place where a loss may occur is where two code bits $T(v_{i_j})_l$ and $T(v_{i_{j+1}})_l$ from the Trevisan code corresponding to two different nodes v_{i_j} and $v_{i_{j+1}}$ are adjacent. However, by the properties of the Trevisan code and because we have arranged the order of the identical groups of strings to be the same as the order $(v_{i_1}, v_{i_2}, \ldots, v_{i_n})$ of the Hamilton path, we are guaranteed that the two copies of the Trevisan code result in at most $2\frac{7}{2}(n-1)n = E$ adjacent but not neighboring zero-one pairs. Therefore, the folding has at most E losses, and one direction has been proved.

2.3 The converse

In this section we summarize the (quite long and involved) proof of the converse; the full proof can be found in the appendix. We consider a folding of the strings with at most E losses; we have to show that it is the intended folding corresponding to a Hamilton path of G.

We define the region R to consist of all points within the $L \times L$ square of the intended folding, as well as all points surrounded by such points. We first prove that the largest component of this region has area at least $L^2 - O(E)$ and perimeter at most $4L + O(E)$ (Lemmas 4 through 8 in the appendix), and therefore the smallest surrounding rectangle has sides of length $L \pm O(\sqrt{LE})$, and there is a

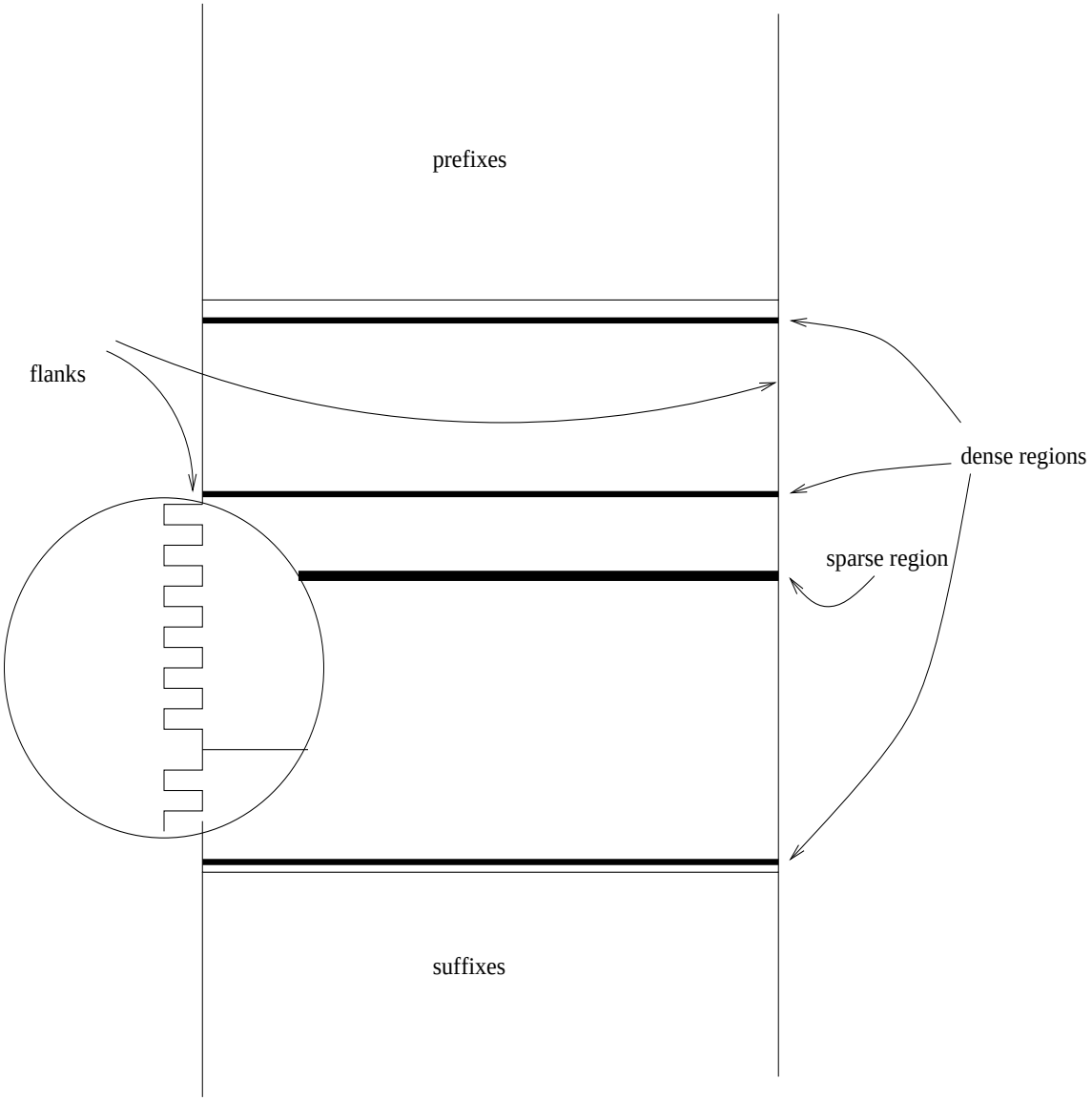


Figure 2: The intended folding.

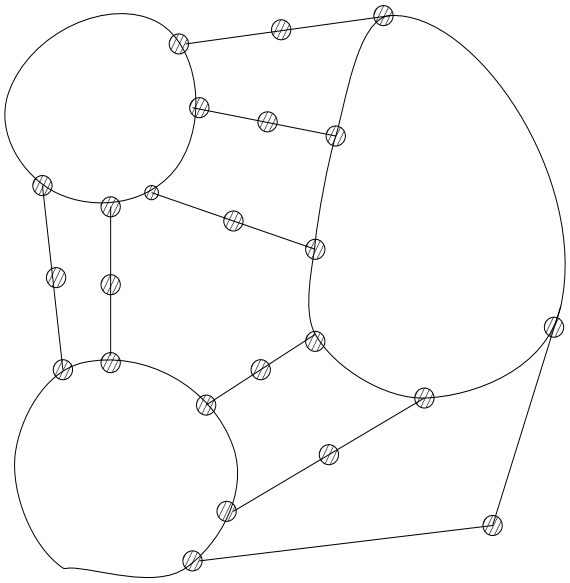


Figure 3: A special planar graph.

square of sides $L - O(\sqrt{LE})$ contained in R (Lemmas 9 and 10). We then consider a string passing through the center of this rectangle, and prove that it is "relatively straight," proceeding without too many bendings, from one end of the square to the opposite (Lemma 11 and Corollary 12). We then prove that any string that passes through a narrow horizontal strip traverses the square from its top to the bottom side, and that in fact that almost all strings so traverse the square (Lemmas 13 and 14 and Corollary 15). It follows that the folding is the intended one, and corresponds to a Hamilton path in G.

3 The string folding problem

In this section we show that the STRING FOLDING PROBLEM (the special case of the multistring problem with $|S| = 1$, which captures the protein folding problem in the 2-dimensional H-P model) is also NP-complete.

Let us call a planar graph *special* if it consists of disjoint faces with nodes of degree three, connected together by paths of length two, and becomes triply connected if all nodes of degree two are collapsed. See Figure 3 for an example.

Theorem 2 *The Hamilton cycle problem remains NP-complete even if restricted to special planar graphs.*

Proof: The reduction from exact cover to planar Hamilton cycle in [7] produces a special planar graph, if the 2-input and 3-input "or" gadgets are replaced by the ones shown in Figure 4. ■

Fix a planar graph G. Two Hamilton cycles are called *orthogonal* if they have the following property:

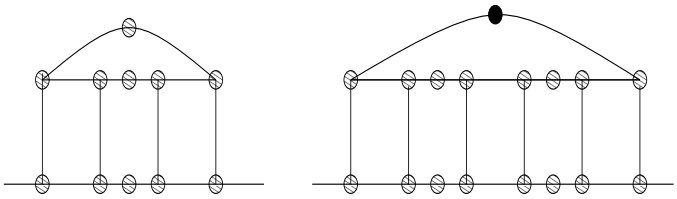


Figure 4: New 2-input and 3-input gadgets.

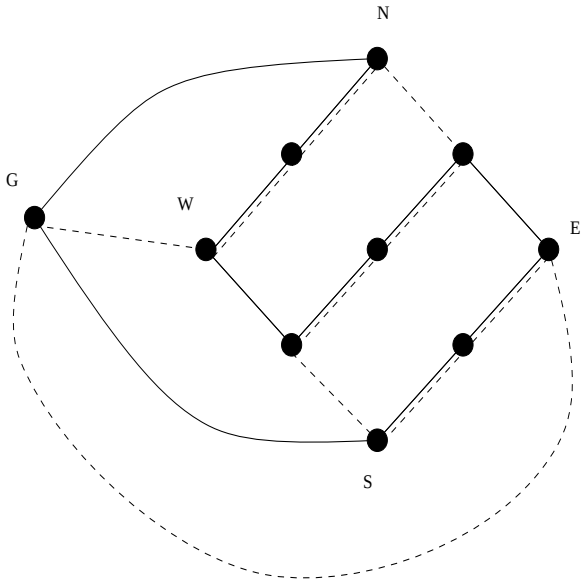


Figure 5: The diamond graph.

Their disjoint union (where we duplicate edges in their intersection) is a degree-4 planar graph with multiple edges which can be embedded in the plane in such a way that the edges around each node alternate between the two cycles. Figure 5 depicts the two Hamilton cycles of the *diamond* graph (plus another node G); they are orthogonal because, by duplicating the three paths of length two, one obtains a degree-four graph around each node of which edges of the two Hamilton cycles alternate.

Suppose that a graph contains the *diamond* graph depicted in Figure 5 (ignore the node G standing for the rest of the graph). The diamond graph has four endpoints, denoted N, S, E, W, whereby it is connected to the rest of the graph. Any Hamilton cycle of the overall graph must traverse the diamond either from N to S, or from E to W (but not, e.g., from E to N).

Theorem 3 *The* STRING FOLDING PROBLEM *is NP-complete.*

Proof: We start from the Hamilton cycle problem for special planar graphs. Given any special planar graph G, we shall modify the graph so that it contains a "standard" Hamilton cycle H_0, such that any Hamilton cycle of the original graph corresponds to a cycle of the modified graph that is orthogonal to H_0. Starting from G and its embedding, take only the degree-2 nodes of G,

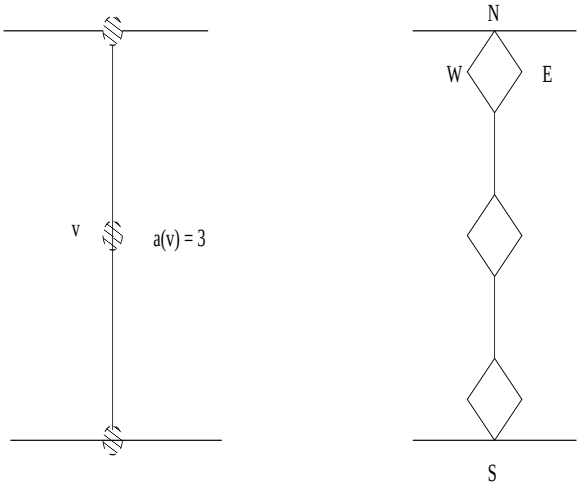


Figure 6: Replacing a degree-2 node by diamonds.

and consider two such nodes adjacent if they are on the same face of the embedding. Since the original graph is special, the resulting graph G' is connected.

Consider thus a cycle C of G' (allowing repeated nodes but no self-loops) that visits all nodes of G' at least once. If the two nodes adjacent to an occurrence of v are on the same face, repeat that occurrence twice. Now for each node v, count the occurrences of v on C and let $a(v)$ be the resulting number.

Replace now each degree-2 node v of G, and its adjacent edges, by $a(v)$ copies of the diamond; the copies are disjoint, and the N and S nodes of the two outermost ones (or the unique one, if $a(v) = 1$) coincide with the nodes of G adjacent to v, see Figure 6. Let $C = (v_0, v_1, \ldots, v_k = v_0)$. For $i = 0, \ldots, k-1$, suppose the ith element of C is the b_ith occurrence of node v_i (let $b_k = 1$); for each $i = 1, \ldots, k$, join the E or W node of the b_{i-1}th copy of the diamond replacing node v_{i-1}, whichever has not been considered up to now, with the E or W node of the b_ith copy of the diamond replacing node v_i, whichever is in the same face with the previous node (for v_0, if $v_1 \neq v_0$, we start with the endpoint, E or W, that is on the same face as v_1, and if $v_1 = v_0$, we start with the endpoint which is not on the same face as v_2). Notice that these new edges do not harm the planarity of the graph, and they, together with the E–W traversal of the diamonds, form the standard Hamilton cycle, H_0, of the resulting graph G''.

H_0 is the only Hamilton cycle of G utilizing a E–W traversal of the diamonds. Any Hamilton cycle utilizing a N–S traversal must correspond to a Hamilton cycle of the original graph G. It is easy to see that any such cycle will be orthogonal to H_0 —because the E–W and the N–S traversals of the diamond are orthogonal.

We shall now construct the instance of the string folding problem. We take any degree-2 node and replace it with two degree-1 nodes, and make these nodes the endpoints of the Hamilton path sought. H_0 becomes a Hamilton path as well. We now perform Trevisan's transformation *having deleted the E–W edges* of the graph (that is, the endpoints of these edges have large Hamming distance in the Trevisan code). We then perform the multistring reduction, with the following

modifications:

- The number of strings corresponding to each city, L/n, is odd. This is trivial to accomplish by adding one string to each set.
- All strings corresponding to the same city are connected together in one string, by ordering them arbitrarily, and connecting the end of the suffix of string $2i-1$ to the end of the suffix of the string $2i$, and the beginning of the prefix of string $2i$ to the beginning of the prefix of string $2i+1$, $i = 1, \ldots, \frac{\frac{L}{n}-1}{2}$.
- Finally, all of these n long strings are connected together in the order suggested by the Hamilton path H_0 by long (of length $2L^4 + 2L^2$) strings of zeros.

We claim that there is a folding with E losses if and only if the original special graph had a Hamilton cycle. Suppose that indeed there is a folding with E or fewer losses. By the precise same argument as in the proof of Theorem 1, there is a Hamilton path in the graph G'' that does not utilize the E–W edges, and hence there is a Hamilton cycle in the original graph.

Conversely, suppose that G did have a Hamilton cycle. Then G'' has a Hamilton cycle H other than H_0, and in fact one that is orthogonal to H_0. But this means that we can arrange the n strings corresponding to the cities as in the intended folding in the proof of Theorem 1, joined together as they are via their prefixes and suffixes in the order suggested by H_0, because H_0 is orthogonal to H. ■

4 Conclusion and further work

Our NP-completeness result settles in the affirmative the widespread conjecture that the protein folding problem, even in its most simple yet realistic two-dimensional H-P simplification, is NP-complete. By a simple modification of our techniques (and, in fact, one that is free of the planarity complications of the present proof) we can show that the three-dimensional version of the protein-folding problem in the H-P model is NP-complete. The appropriate modification of our proof is this: The string consists of roughly $L \times L \times L$ ones, which form a cube protected from all sides (except for the eight corners, where 8 losses *must* occur) by zeros. The Hamilton cycle problem is encoded in the part of the cube that lies just below one particular face of the cube. This part is traversed in one direction by "tubes" whose cross-section is a square of four zeros. Mismatches in the alignment of these tubes correspond to mismatches in the Trevisan code of the underlying graph, and contribute the only extra losses, beyond the eight mandatory ones. Thus the intended folding has a total of $4E + 8$ losses, and the steps in establishing the converse are analogous to the ones in the present proof. The same result was proven independently, and a few months earlier, by Berger and Leighton [1].

References

[1] B. Berger, F. T. Leighton, manuscript submitted to *J. Mol. Biol.*, July 1997.

[2] H. S. Chan, K. A. Dill. The protein folding problem. *Physics Today* (1993), pp. 24-32.

[3] K. A. Dill. Dominant forces in protein folding. *Biochemistry* **29** (1990), pp. 7133-7155.

[4] K. A. Dill, S. Bromberg, K. Yue, K. Fiebig, D. P. Yee, P. D. Thomas, H. S. Chan. Principles of protein folding - A perspective from simple exact models. *Protein Science* **4** (1995), pp. 561-602.

[5] A. S. Fraenkel. Complexity of protein folding. *Bulletin of Mathematical Biology* **55** (1993), pp. 1199-1210.

[6] W. E. Hart, S. Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal. *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing*, 1995, pp. 157-168.

[7] D. S. Johnson, C. H. Papadimitriou. "Computational Complexity," in *The Traveling Salesman Problem,* edited by E. L. Lawler, J. K. Lenstra, A. H. G. Rinooy Kan, D. B. Shmoys, Wiley Interscience, 1985.

[8] J. King. Deciphering the rules of protein folding. *Chemical Engineering News* **67** (1989), pp. 32-54.

[9] A. Nayak, A. Sinclair, U. Zwick. Spatial Codes and the Hardness of String Folding. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, to appear.

[10] J. T. Ngo, J. Marks, M. Karplus. Computational complexity, protein structure prediction, and the Levinthal paradox. In *The protein folding problem and tertiary structure prediction,* edited by K. M. Merz and S. M. Le Grand, Birkhauser, Boston, 1994.

[11] M. Paterson, T. Przytycka. On the complexity of string folding. *Discrete Applied Mathematics* **71** (1996), pp. 217-230.

[12] L. Trevisan. When Hamming Meets Euclid: The Approximability of Geometric TSP and MST [Extended Abstract]. *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, 1997, pp.21-29.

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

A The converse

In this section we prove that if there is a folding of the strings in S with at most E losses, then there is a Hamilton path from v_1 to v_n in G. Let f be a folding of the strings in Z^2 such that the resulting number of losses is less than or equal to E. Since the embedding is injective, we will identify bits in the strings with their images in Z^2, and call a point in Z^2 a *zero* or a *one* if it has a preimage which is, respectively, a zero or a one; otherwise, the point will be called *empty*.

Consider the region R in Z^2 which consists of all internal points (points in the $L \times L$ square in the intended folding) as well as all points surrounded by a discrete closed curve of internal points, where, for the purposes of a discrete curve, two points (x_1, y_1) and (x_2, y_2) in Z^2 may be joined if they are adjacent (joined by a vertical or horizontal edge in the lattice Z^2), or if $|x_1 - x_2| = 1$ and $|y_1 - y_2| = 1$ —that is, we allow diagonal edges. We include points surrounded by a discrete curve of internal points in R so that R will not contain any "holes." Let R_C be the largest connected subset of R, where for connectivity we allow only vertical or horizontal edges. We will prove that R_C looks very much like an $L \times L$ square and that, for the most part, strings passing through R_C are approximately straight and parallel.

It will be helpful to visualize R not only as a collection of points but also as a collection of continuous regions. By visualizing each point contained in R as a unit square centered at the point and parallel to the axes, then subsets of R may be said to have perimeter and area, denoted by the functions $Perim()$ and $Area()$. Our immediate goal will be to prove lemmas bounding the perimeter and area of R_C.

We begin by bounding the perimeter. Define the boundary of a region A, denoted $Bdary(A)$, to consist of all points in A adjacent to at least one point not in A. Then the perimeter of any subset of R consisting of connected components of R is equal to the number of boundary points it contains plus the number of convex corners formed by points on the boundary of the region. We make a few further definitions before we state our first lemma. We call an internal point *straight* if its two neighbors lie in the same vertical or horizontal line; otherwise, the point is *bent*. A point $q \in Z^2$ is said to be *within distance d (vertical distance d)* of a point $p \in Z^2$ if q is reachable from p using a path of at most d vertical and horizontal edges (at most d vertical edges and any number of horizontal edges). A point $p \in Z^2$ is said to have a *loss within distance d (vertical distance d)* if there exists a loss involving two points within distance d (vertical distance d) of p. Finally, a set of points is said to *have a loss within distance d (vertical distance d)* if it contains a point which has a loss within distance d (vertical distance d). The first lemma follows:

Lemma 4 *There exists a positive constant c_1 such that*

$$|Bdary(R_C)| \leq |Bdary(R)| \leq 4L + c_1 E - 4.$$

Proof: Since any boundary point of R_C is also a boundary point of R, the first inequality is clear. To prove the second inequality, we first note that all points on the boundary of R must be internal points; for a point of R which is not internal must be surrounded by a curve of internal points and hence cannot be on the boundary. Suppose, then, that there is an internal point, p, on the boundary of R which is not one of the $4L - 4$ intended boundary points. We prove that there

must either be a loss within distance 6 of p or an intended corner within distance 2 of p. There are three cases to consider:

Case 1: p is a one.

Point p must be adjacent to a point, q, which is not an internal point. If q is empty, then there is a loss within distance one. If q is a zero, then there must also be a loss within distance one because no one which is not on the intended boundary has a non-internal zero as a neighbor in any of the strings.

Case 2: p is a straight internal zero.

Since p is a non-flank zero and there is no loss within distance 6, the picture of the neighborhood around p (up to rotation) must be as follows:

$$\begin{array}{c} 1 \;\_\; 1 \\ 1{-}p{-}1, \end{array}$$

where the isolated pipe represents the region boundary. However, the two possibilities for the point between the two uppermost ones, empty or a non-internal zero, both lead to losses within distance two of p.

Case 3: p is a bent internal zero.

Since p is on the boundary and, thus, cannot be adjacent to two other internal zeros, it follows from Lemma 5, stated below, that p must have a loss within distance 6 or an intended corner within distance 2.

We have completed showing that there must either be a loss within distance 6 of p or an intended corner within distance 2. Since there are at most a constant number of points within distance 6 of the two points involved in a loss and there are at most E points (for $n \geq 3$) within distance 2 of an intended corner, we can set c_1 equal to the constant plus four and the lemma follows. ■

We next state a lemma involving non-flank internal zeros referred to above.

Lemma 5 *If z is a bent non-flank internal zero which is not adjacent to two other non-flank internal zeros nor within distance 2 of an intended corner, then there must be a loss within distance 6 of z.*

Proof: Suppose that there is a bent non-flank internal zero, z, which is not adjacent to two other non-flank internal zeros nor within distance 2 of an intended corner. The assumption of no intended corners close to z implies that if there is a one within distance 2 of z which is contained in the substring 010, then it must either be contained in the larger substring 1010 or 0101. Suppose, in addition, that there is no loss within distance 6 of z. We show that in every potential configuration involving z these assumptions lead to a contradiction, for there must, in fact, be a loss within distance 6 of z.

Assuming there is no loss within distance 6 of z, the picture of its neighborhood under folding f (up to rotation) must be as follows:

$$\begin{array}{cccc} & 0 & -1 & \\ 0 & z & -1 & 1 \\ | & | & & \\ 1 & 1. & & \end{array}$$

The reason for the appearance of the ones other than the two neighbors of z is that there are always at least eight ones which lie between internal zeros on any string, flank or non-flank (though on flanks they may be separated by some groups of 4 zeros). The same separation is true for internal zeros and prefix or suffix zeros. This fact will play a role in almost all of the pictures of configurations which follow in the proof of this lemma and in the proof of Lemma 6.

Next, we consider the classifications of the zeros lying adjacent to z. Note that the two zeros cannot both be straight.

Case 1: One of the adjacent zeros is a bent flank internal zero:

Assume without loss of generality that it is the upper adjacent zero. Since there are no losses within distance 6, the picture must be as follows:

$$\begin{array}{cccc} & 1 & X & \\ & | & & \\ & 0 & -1^* & \\ & & & X \\ 0 & z & -1 & 1 \\ | & | & & \\ 1 & 1. & & \end{array}$$

No matter where the zero following the starred one is placed, there will be a loss (located at one of the two X's), a contradiction.

Case 2: One of the adjacent zeros is a bent non-flank internal zero and the other adjacent zero is not a bent non-internal zero.

Assume without loss of generality the bent non-flank internal zero is the upper adjacent zero. The necessary picture follows:

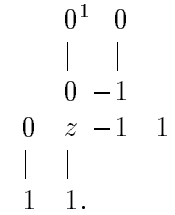
$$\begin{array}{cccc} 1 & 1 & & \\ | & | & & \\ 0 & 0 & -1 & \\ 0 & z & -1 & 1 \\ | & | & & \\ 1 & 1. & & \end{array}$$

It should be clear that the adjacent zero to the left of z cannot be straight (there are no maximal substrings of zeros of length two). Since we assumed it was not a bent non-internal zero and it is not a bent internal zero, this case is impossible.

Only one case remains:

Case 3: One of the adjacent zeros is a bent non-internal zero.

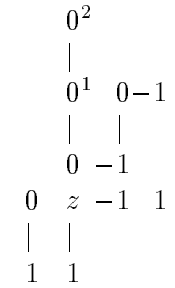
Assume it is the upper zero. The picture must be the following:



¿From the picture it should be clear that the upper zero adjacent to z cannot be a prefix or suffix zero due to our assumption about intended corners. It must be contained in one of the groups of 4 zeros on a flank. As well, the zero which is a neighbor to the 1-neighbor of the upper zero may also not be a prefix or suffix zero.

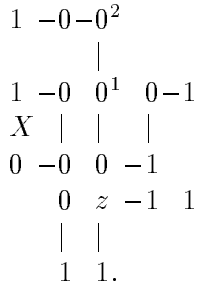
We next examine the potential locations for the zero following the zero marked with the superscript one (zero 1).

Case 3A: The zero following zero 1 goes up, as appears below.



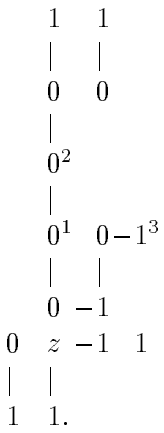
We examine the potential locations for the zero following the zero marked with the superscript two (zero 2).

Case 3Ai: The zero following zero 2 goes left:

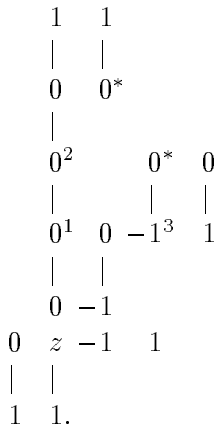


There must be a loss at the X.

Case 3Aii: The zero following zero 2 goes up:

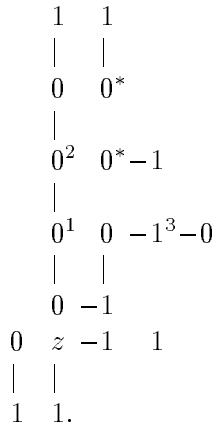


Case 3Aiia: The zero following one 3 goes up:



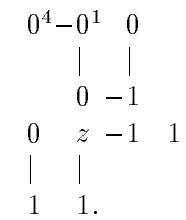
This configuration is impossible because no matter how the picture is completed, the two starred zeros must belong to the same string, but no string contains either a maximal substring of zeros of length three or the substring 10101.

Case 3Aiib: The zero following one 3 goes right:

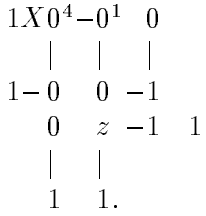


Similarly to the previous subcase, this configuration is impossible because the two starred zeros must be in the same string, but no string contains a maximal substring of zeros of length two.

Case 3B: The zero following zero 1 goes left:

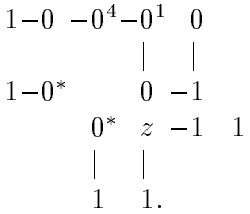


Case 3Bi: The zero following zero 4 goes down:



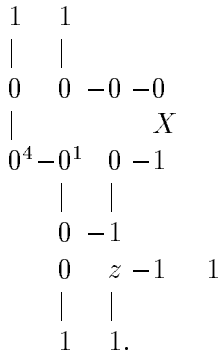
There must be a loss at the X.

Case 3Bii: The zero following zero 4 goes left:



Here, again, the configuration is impossible because the two starred zeros must be contained in the same string in a maximal substring of zeros of length three.

Case 3Biii: The zero following zero 4 goes up:



There must be a loss at the X, contradiction.

We have completed our case analysis. In all cases, we showed that under our assumptions the potential configurations must, in fact, either be impossible or contain a loss within distance 6 of z (all points shown in the figures were within distance 6 of z). Thus, a bent non-flank internal zero which is not adjacent to two other non-flank internal zeros nor within distance 2 of an intended corner must have a loss within distance 6. ■

The following lemma, which is the final step toward limiting $Perim(R_C)$, bounds the number of concave corners on the boundary of any subset of R.

Lemma 6 *If there is a concave corner on the boundary of R, then there must either be a loss within distance 7 of the point of R at the corner or an intended corner within distance 4 of the corner point. Consequently, there exists a positive constant, c_2, such that there are at most $c_2 E$ concave corners on the boundary of any subset consisting of connected components of R.*

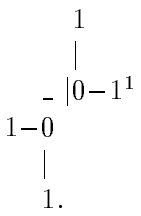
Proof: Suppose there is a concave corner on the boundary of R such that no intended corner is within distance 4 of the corner point and there are no losses within distance 7 of the corner point. As in the proof of Lemma 5, the assumption of no nearby intended corner implies any one within distance 4 of the corner point which is contained in the substring 010 must be contained in the larger substring 1010 or 0101. Also similarly to the proof of Lemma 5, we show these assumptions must lead to a contradiction.

We perform a case analysis based on the classification (zero or one) of the sides of the the concave corner. Note that no bent non-flank internal zero may be a side of the concave corner, for since there can be no intended corner in distance 2 of the zero (and it is not adjacent to two non-flank internal zeros), there must be a loss within distance 6 of the zero. However, this implies the loss is within distance 7 of the corner point.

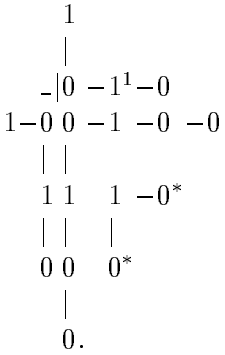
Case 1: Suppose two ones form the sides of the concave corner. We represent this configuration using the following diagram:

The two pipes represent where the ones meet the perimeter of R. The two possibilities for the classification of the point not in R which is adjacent to the two ones, either empty or a non-internal zero, both lead to a loss within distance 2 of the corner point, contradiction.

Case 2: Two zeros form the sides of the concave corner. Since there is no loss within distance 7 of the corner point, the necessary picture is the following:

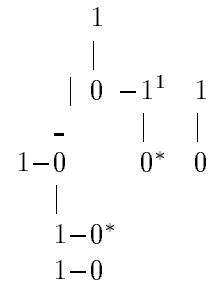


Case 2A: The zero following one 1 goes right:



One of the starred zeros must be followed by a one, so there must be a loss within distance 4 of the corner point, contradiction.

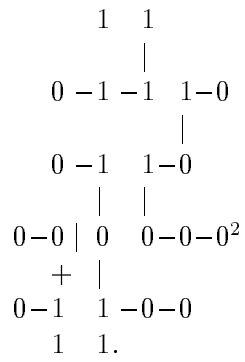
Case 2B: The zero following one 1 goes down:



Note the starred zero not adjacent to one 1 must go to the right as shown, for otherwise there would be a loss as in Case 2A. One of the starred zeros must be bent, but then there would be a maximal substring of zeros of length two. Thus the configuration is impossible.

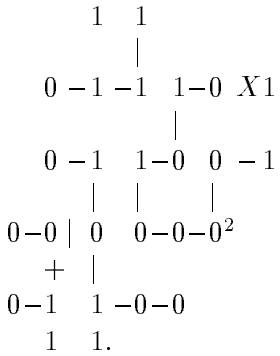
Case 3: A zero and a one form the sides of the concave corner.

Case 3A: Suppose the zero is straight. The necessary picture follows:



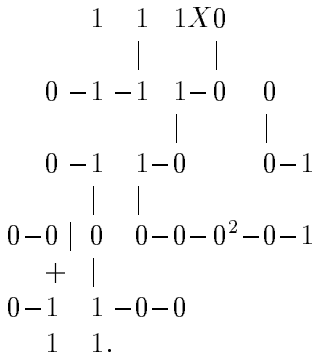
The plus sign denotes the perimeter as well as the link that crosses it.

Case 3Ai: The zero following zero 2 is up:



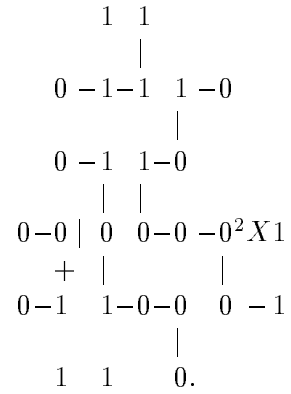
There must be a loss at the X.

Case 3Aii: The zero following zero 2 is right:



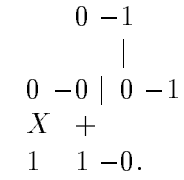
Again, there must be a loss at the X.

Case 3Aiii: The zero following zero 2 is down:



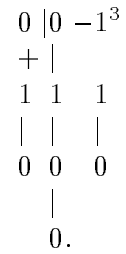
Again, there must be a loss.

Case 3B: Suppose the side zero is bent so that its neighbors go up and to the right. The necessary picture follows:

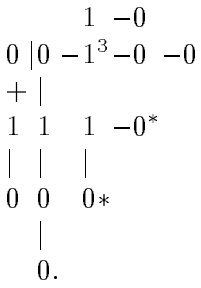


There must be a loss at the X.

Case 3C: Suppose, finally, the side zero is bent so that its neighbors go right and down:

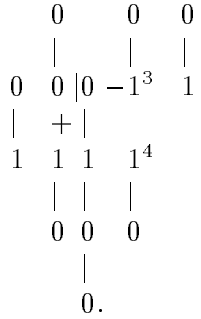


Case 3Ci: The zero following one 3 goes right:

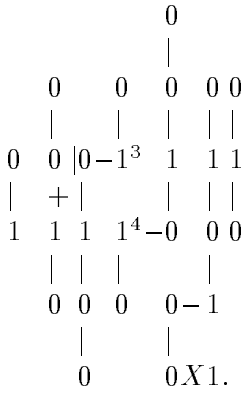


There must be a loss within distance 4 of the corner point due to the one which must follow one of the starred zeros.

Case 3Cii: The zero following one 3 goes up:

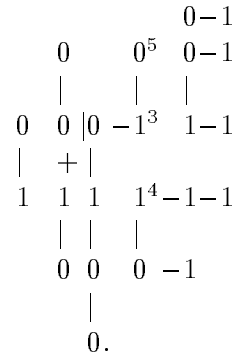


Case 3Ciia: The bit following one 4 is a zero:

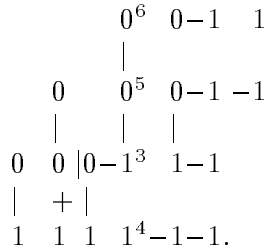


There must be a loss at the X.

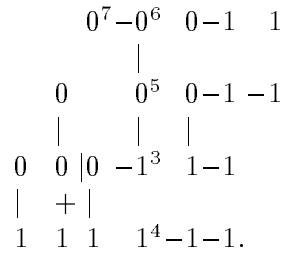
Case 3Ciib: The bit following one 4 is a one:



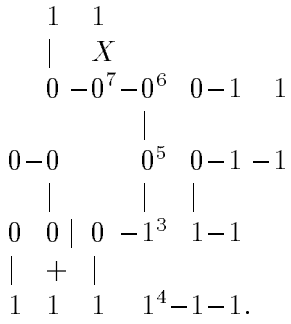
Case 3Ciib1: The zero following zero 5 is up:



Case 3Ciib1A: The zero following zero 6 is left:

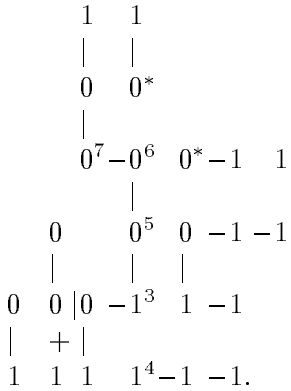


Case 3Ciib1Ai: The zero following zero 7 is left:



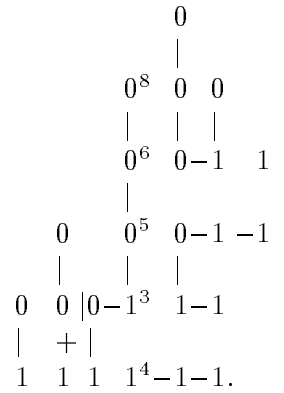
There must be a loss at the X.

Case 3Ciib1Aii: The zero following zero 7 is up:

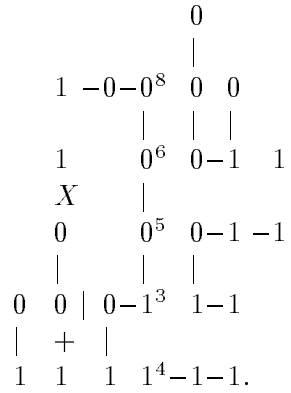


The two starred zeros must lie in the same string, but no string contains either a maximal substring of zeros of length three or the substring 10101. Thus, the configuration is impossible.

Case 3Ciib1B: The zero following zero 6 is up:

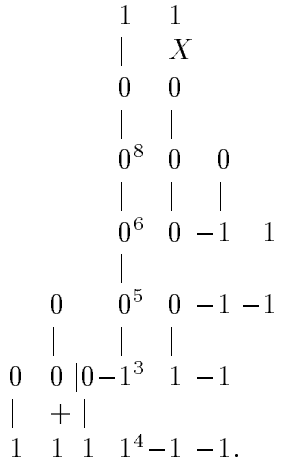


Case 3Ciib1Bi: The zero following zero 8 is left:



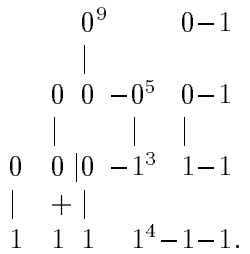
There must be a loss at the X.

Case 3Ciib1Bii: The zero following zero 8 is up:

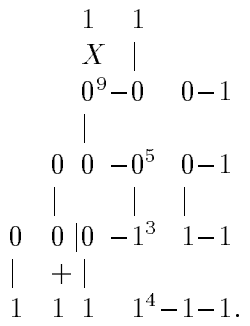


There must be a loss at the X.

Case 3Ciib2: The zero following zero 5 is left:

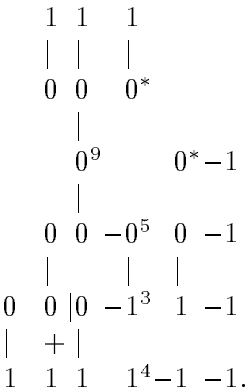


Case 3Ciib2A: The zero following zero 9 is right:



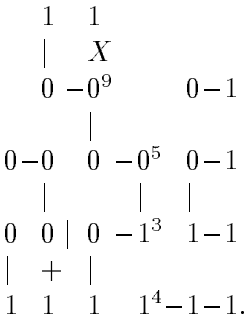
There must be a loss at the X.

Case 3Ciib2B: The zero following zero 9 is up:



This configuration is impossible because the two starred zeros must belong to the same string, but no string contains a maximal substring of zeros of length three (or the substring 10101).

Case 3Ciib2C: The zero following zero 9 is left:



Again, we find a loss, a contradiction.

This completes our case analysis. In all cases, we showed that under our assumptions the potential configurations must either be impossible or contain a loss within distance 7 of the corner point (all points shown in the figures were within distance 7 of the corner point). Consequently, a concave corner on the boundary without an intended corner within distance 4 of the corner point must have a loss within distance 7 of the corner point. Since there are at most E points within distance 4 of an intended corner (for $n \geq 4$) and there are at most a constant number of points within distance 7 of the two points involved in a loss, setting c_2 equal to the constant plus four, we conclude that there are at most $c_2 E$ concave corners on the boundary of R. Finally, any subset consisting of connected components of R has at most as many concave corners as R. This completes the proof of the lemma. ■

Corollary 7 *There exists a positive constant, c_3, such that*

$$Perim(R_C) \leq 4L + c_3 E.$$

Proof: Note that the number of convex corners contained in R_C is four plus the number of concave corners. The corollary, then, follows from the observation that the perimeter of R_C is equal to the number of boundary points plus the number of convex corners it contains. Using Lemmas 4 and 6, we may take $c_3 = c_1 + c_2$. ■

We now prove a lower bound on $Area(R_C)$ and, in particular, we prove that, by limiting ourselves to the connected region R_C, we do not lose many internal points.

Lemma 8 *There exists a positive constant, c_4, such that*

$$Area(R_C) \geq L^2 - c_4 E.$$

More specifically, R_C contains all the internal substrings of the non-flank strings and all but at most $c_4 E$ internal points contained in the flanks.

Proof: Let R_N consist of the connected components of R which contain the internal points of the $L - 2$ non-flank strings. Since all the internal points from a non-flank string must lie in the same connected component, each component of R_N must contain at least L points, and there

are at most $L-2$ such components. We show $R_N \subseteq R_C$ by proving that, in fact, R_N contains only one connected component. Suppose R_N consists of $x \leq L-2$ connected components, for a positive integer x. First, we find a lower bound on $Perim(R_N)$ by making use of the fact that any component of area A must have perimeter greater than or equal to $4\sqrt{A}$. Using this inequality to find a lower bound on the perimeter, the limiting situation is when $x-1$ components are as small as possible and the remaining points are contained in the last component. A lower bound on $Perim(R_N)$ is, thus, given by $4(x-1)\sqrt{L} + 4\sqrt{L^2-(x+1)L}$. On the other hand, we can find an upper bound on $Perim(R_N)$ by finding upper bounds on $|Bdary(R_N)|$ and the number of convex corners contained in R_N. Clearly $|Bdary(R_N)| \leq |Bdary(R)|$, since any boundary point of R_N is also a boundary point of R, and the number of convex corners contained in R_N is at most $4x$ plus the number of concave corners contained in R_N. Therefore, using Lemmas 4 and 6, we see that $4L + c_1E - 4 + 4x + c_2E$ is an upper bound on $Perim(R_N)$. Comparing upper and lower bounds, it must be true that

$$\begin{aligned} 4x + 4L + (c_1+c_2)E - 4 &\geq 4(x-1)\sqrt{L} + 4\sqrt{L^2-(x+1)L} \\ &\geq 4(x-1)\sqrt{L} + 4(L-(x+1)) \\ &= (4\sqrt{L}-4)x + 4L - 4\sqrt{L} - 4. \end{aligned}$$

Consequently, $(4\sqrt{L}-8)x \leq 4\sqrt{L} + (c_1+c_2)E$. Since $4\sqrt{L} - 16 > (c_1+c_2)E$ for $n \geq 10$, we must have $x = 1$. Therefore, since R_C is the largest connected component of R, we must have $R_N \subseteq R_C$, meaning the internal substrings of the non-flank strings are contained in R_C, and thus $Area(R_C) \geq L^2 - 2L$.

Suppose more than $(c_3+1)E$ internal flank points are not contained in R_C. Then $perim(R_C) < 4L + c_3E - (c_3+1)E = 4L - E$ since we have lost the use of more than $(c_3+1)E$ intended boundary points. However, we still must have

$$\begin{aligned} Perim(R_C) &\geq 4\sqrt{L^2-2L} \\ &\geq 4L - 8, \end{aligned}$$

which leads to a contradiction since $E \geq 8$ for $n \geq 2$. The lemma follows by taking $c_4 = c_3 + 1$. ■

We next show that R_C is, in fact, very similar to an $L \times L$ square by bounding the dimensions and area of the smallest rectangle containing R_C and the dimensions of a square strictly contained in R_C.

Lemma 9 *The smallest rectangle containing R_C has sides of length $L+a$ and $L+b$, where $c_4\sqrt{LE} \geq a, b \geq -c_4\sqrt{LE}$. Further, its area is at most $L^2 + c_5LE$, where c_5 is a positive constant.*

Proof: Let $a = \max\{|x_1 - x_2| + 1 : (x_1,y_1),(x_2,y_2) \in R_C\} - L$ and let $b = \max\{|y_1 - y_2| + 1 : (x_1,y_1),(x_2,y_2) \in R_C\} - L$. From the definitions it should be clear that the smallest rectangle containing R_C has sides of length exactly $L+a$ and $L+b$. Comparing the area and perimeter of the bounding rectangle to those of (the continuous version of) R_C, we derive the following inequalities:

1. $(L+a)(L+b) \geq L^2 - c_4E \Rightarrow (a+b)L + ab \geq -c_4E$.

2. $2(L+a+L+b) \leq 4L + c_3E \Rightarrow a+b \leq \frac{c_3}{2}E$.

If a and b are both positive or both negative, and since $c_4 > c_3$, it is certainly true that $c_4E \geq a, b \geq -c_4E$. We note for later use that $ab \leq c_4^2E^2$, regardless of the signs of a and b. Now let us assume without loss of generality that a is positive and b negative. Substituting bounds for $a+b$ and b in inequality 1 using inequality 2, we derive:

$$\frac{c_3}{2}EL + a(\frac{c_3}{2}E - a) \geq -c_4E \Rightarrow a(a - \frac{c_3}{2}E) \leq \frac{c_3}{2}LE + c_4E.$$

Since $(\frac{c_3}{2}\sqrt{LE} + \frac{c_3}{2}E)(\frac{c_3}{2}\sqrt{LE}) = \frac{c_3^2}{4}LE + \frac{c_3^2}{4}E\sqrt{LE} \geq \frac{c_3}{2}LE + c_4E$, we must have $a \leq \frac{c_3}{2}\sqrt{LE} + \frac{c_3}{2}E \leq c_4\sqrt{LE}$, which implies $b \geq -c_4\sqrt{LE}$ using inequality 1.

To verify the bound on the area of the rectangle, we note that $(L+a)(L+b) = L^2 + (a+b)L + ab \leq L^2 + \frac{c_3}{2}LE + c_4^2E^2$ using inequality 2 and the bound on ab given above. Taking $c_5 = \frac{c_3}{2} + c_4^2$, the bound stated in the lemma follows. ■

We may now state a lemma which finds a large square strictly contained in R_C.

Lemma 10 *For a positive constant c_6, there exists a square with sides of length $L - c_6\sqrt{LE}$ contained in R_C.*

Proof: We proved in the previous lemma that the smallest rectangle containing R_C has area less than or equal to $L^2 + c_5LE$, for a constant c_5. Since R_C contains at least $L^2 - c_4E$ internal points by Lemma 8, clearly any region contained in the bounding rectangle of area $2c_5LE$ must contain an internal point contained in R_C.

At this point, it will help to have the bounding rectangle and R_C oriented in the coordinate plane. Let us assume without loss of generality that one of the up to 4 potential central points of the bounding rectangle is (0,0). (This is just a translation of the the embedding f.) Now consider the square, $SQ1$, with sides of length $L - c_4\sqrt{LE}$ which is centered at (0,0). Since $SQ1$ is contained in the bounding rectangle we know that if we consider the 4 squares with sides of length $\sqrt{2c_5LE}$ (and area $2c_5LE$) which share a corner with $SQ1$, then there must be a point of R_C contained in each of those 4 squares. That is, there exist 4 points, $(x_1, y_1), (x_2, y_2), (x_3, y_3)$, and (x_4, y_4), in R_C such that, for each $i \in [4]$, (x_i, y_i) is in quadrant i and $|x_i|, |y_i| \geq \frac{1}{2}(L - c_4\sqrt{LE}) - \sqrt{2c_5LE}$. $SQ1$ and the 4 points are pictured in Figure 7.

Consider the square, $SQ2$ (also pictured in Figure 7), which has sides of length $L - (c_4 + 2\sqrt{2c_5})\sqrt{LE}$ and is centered at the origin. Since R_C is connected and we have found the points $(x_i, y_i) \in R_C$, $i \in [4]$, we must have that $Perim(R_C) \geq Perim(SQ2) = 4L - 4(c_4 + 2\sqrt{2c_5})\sqrt{LE}$. Moreover, if there exists a point in the interior of $SQ2$ but not in R_C which is at a distance of more than $4(c_4+2\sqrt{2c_5})\sqrt{LE}+c_3E$ from all the sides of $SQ2$, then, since there are no holes in R_C, we must have $Perim(R_C) > 4L+c_3E$. This is clearly impossible due to Corollary 7. Thus, any square with sides of length less than $L-(c_4+2\sqrt{2c_5})\sqrt{LE}-2(4(c_4+2\sqrt{2c_5})\sqrt{LE}+c_3E) = L-(9c_4+18\sqrt{2c_5})\sqrt{LE}-2c_3E$ centered at the origin must be contained in R_C. In particular, the square, SQ, centered at (0,0) with sides of length $L - c_6\sqrt{LE}$, where $c_6 = 29c_5$, is contained in R_C. ■

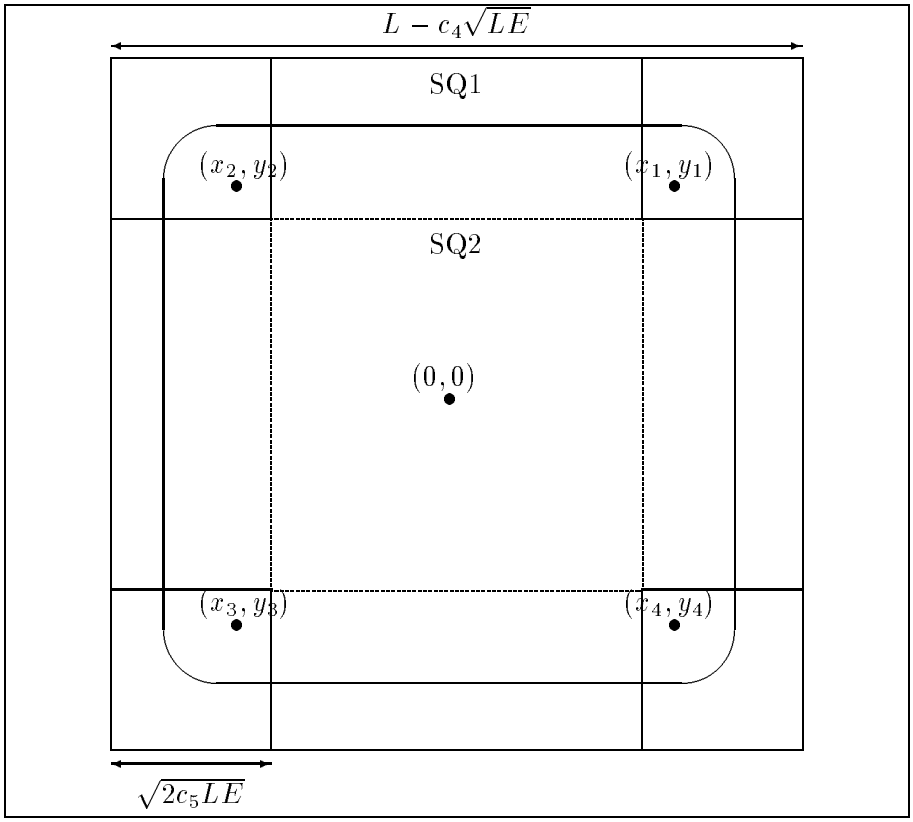


Figure 7: $SQ1, SQ2$, and 4 points of R_C.

We now move closer to our goal of showing that many strings pass through R_C in an approximately straight and parallel fashion by first showing that there exists a non-flank string passing near the origin which is approximately straight. Note that it is impossible for any prefix or suffix zero to be in R_C since that would imply the whole prefix or suffix is in R_C, forcing its perimeter to be too high. Thus the string must exit R_C and, in particular, SQ, at two points. We show the two points at which the string near $(0,0)$ exits SQ (not to return) are near the centers of two opposite sides of SQ. As a corollary to the placement of this string, we derive that the flanks must lie at opposite sides of R_C.

Lemma 11 *There exists a non-flank string passing within an $O(\sqrt{LE})$ distance of $(0,0)$ which exits SQ within $O(\sqrt{LE})$ distance of the centers of two opposite sides. Since a side of SQ has length $L - c_6\sqrt{LE}$, the string must then be relatively straight, containing $O(\sqrt{LE})$ bent points.*

Proof: It should be clear that the second sentence follows from the first since the internal substring of a non-flank string has length L. Of the L bits, $L - c_6\sqrt{LE}$ must be used to get from one side of the square to the other, leaving a slack of size only $c_6\sqrt{LE}$.

Consider the square with side length $\sqrt{2c_5LE}$ which is centered at the origin. This square must contain an internal point, p, of R_C. The string, s, containing p cannot be a flank because we would lose the use of too many intended boundary points inside SQ, so we have found a non-flank string passing near (0,0). Point p is at a distance of at least $\frac{1}{2}(L - c_6\sqrt{LE}) - \sqrt{2c_5LE}$ from each side of SQ. Thus s cannot exit SQ at any point farther than $(c_6 + 3\sqrt{2c_5})\sqrt{LE}$ from the centers of the sides of SQ.

There are then 3 possibilities for the way in which string s exits SQ: it either exits at a single side (forms a U-shape), exits at adjacent sides (forms an L-shape), or exits at opposite sides as desired. We prove the first two cases are impossible. Suppose, first, that s forms a U-shape. For purposes of illustration we'll assume s exits through the top side of SQ. Recall the structure of the non-flank strings: there are 3 dense substrings of length $\frac{L}{10}$, the middle of which begins with bit number $\frac{3L}{10} + 1$ of the internal substring. We claim that no matter how s is folded into a U-shape, the bits on the opposite side of the U, for illustration, the right side, facing bits $\frac{3L}{10} + 1$ through $\frac{2L}{5}$ on the left must be ones. The string must have at least $L - (c_6 + 2\sqrt{2c_5})\sqrt{LE}$ bits inside SQ, and by evaluating the two limiting configurations of the U (where the at most $(c_6 + 2\sqrt{2c_5})\sqrt{LE}$ bits are outside SQ), we see that bits $\frac{3L}{10} + 1$ through $\frac{2L}{5}$ may only be opposite bits in the range

$$[\frac{7L}{10} + 1 - (c_6 + 2\sqrt{2c_5})\sqrt{LE}, \frac{4L}{5} + (c_6 + 2\sqrt{2c_5})\sqrt{LE}].$$

We may assume $(c_6 + 2\sqrt{2c_5})\sqrt{LE} \leq \frac{L}{180} - E - 1$, which is true for $n \geq 10$. Clearly, then, $(c_6 + 2\sqrt{2c_5})\sqrt{LE} \ll L/10$ and the bits in the above range are ones. Next, again because the string has slack only at most $(c_6 + 2\sqrt{2c_5})\sqrt{LE}$, at most that many zeros out of the $\frac{L}{90}$ in the middle dense substring may be horizontal or bent. Thus we may assume there are at least $E + 1$ vertical zeros. Now, if there is no loss in the window containing a vertical zero and its neighbors and the three points adjacent to them on the right, then the point adjacent to the vertical zero on the right must also be a vertical zero. Assuming the adjacent vertical zero is also a non-flank zero without a loss

in a similar window, then the pattern will continue and there will be a horizontal line of vertical zeros:

$$\begin{array}{ccc} 1 & 1 & 1 \\ | & | & | \\ 0 & 0 & 0\cdots \\ | & | & | \\ 1 & 1 & 1. \end{array}$$

Thus, there are $E+1$ lines of vertical zeros extending from the left side of the U toward the right which may not be terminated unless they reach a flank, impossible due to the resulting loss of intended boundary points, or unless one of the zeros in the line has a loss within its window containing six points. Since a loss terminating a line may be uniquely associated with that line, one of the $E+1$ lines of zeros must have no associated losses. However, this line of zeros must run into a one on the right side of the U, resulting in a loss and a contradiction. We conclude the string cannot form a U-shape.

Let us, next, suppose string s forms an L-shape. We prove this is impossible by using a similar argument involving lines of zeros. One half of the L (for illustration we assume it exits through the top side of SQ) contains the middle, dense substring and, as above, we are guaranteed there is at least one horizontal line of vertical zeros without an associated loss in distance two. Consider the horizontal second half of the L, say exiting through the right side of SQ, which contains the rightmost dense substring. This second half contains a portion of at least $\frac{L}{10}-(c_6+2\sqrt{2c_5})\sqrt{LE} \geq \frac{3L}{40}$ of the rightmost dense substring, and since at most $(c_6+2\sqrt{2c_5})\sqrt{LE}$ of the first $\frac{L}{180}$ zeros in the dense substring are vertical or bent, at least $E+1$ zeros are horizontal (and at a distance at least $\frac{L}{40}$ from the right side of SQ). These zeros must form vertical lines of horizontal zeros, and, using a symmetrical argument, we are guaranteed that at least one of the lines will be without an associated loss in distance two. However, this line must encounter the horizontal line of vertical zeros originating from the first half of the L, and no flank may terminate either of these lines between the sides of the L and the point where they meet. Since the straight zeros in the lines are surrounded by ones, one of the two lines must suffer a loss, a contradiction. Therefore, the string cannot form an L. ■

We may assume without loss of generality that string s exits at the top and bottom of SQ and thus its orientation is vertical. (This is just a rotation of the folding f.) We may now place the flanks:

Corollary 12 *The flanks both intersect the lines* $y = \pm(\frac{1}{2}(L - c_6\sqrt{LE}) - c_7\sqrt{LE})$, *where* c_7 *is a positive constant. Further, in the vertical interval stretching between the two lines, one flank lies to the left of and one flank lies to right of the lines* $x = \pm(\frac{1}{2}(L - c_6\sqrt{LE}) - \frac{3}{2}c_4E)$.

Proof: We use horizontal lines of vertical zeros to place the flanks. In fact, the only way for a horizontal line of vertical zeros to end without a loss in vertical distance one is for it to end in a vertical flank zero:

$$
\begin{array}{ccc}
1 & 1 & 1-0 \\
| & | & | \\
0 & 0 & 0 \\
| & | & | \\
1 & 1 & 1-0.
\end{array}
$$

Using the first and last dense substrings contained in the vertical string, we can find horizontal lines of vertical zeros with no associated losses at distances of no more than $8c_6\sqrt{LE}+9(E+1) \leq c_7\sqrt{LE}$, where $c_7 = 17c_6$, from the top and bottom of SQ. The central string s prohibits the flanks from stretching across the top or bottom of R_C to complete these lines; the prefix and/or suffix of s would cause the loss of too many intended boundary points. Consequently, one flank must complete both of the lines on the left and the other flank must complete both of the lines on the right. Thus, both flanks intersect the lines $y = \pm(\frac{1}{2}(L - c_6\sqrt{LE}) - c_7\sqrt{LE})$.

Suppose that in the vertical interval $[-\frac{1}{2}(L - c_6\sqrt{LE}) + c_7\sqrt{LE}), \frac{1}{2}(L - c_6\sqrt{LE}) - c_7\sqrt{LE}]$ a flank is further than $\frac{3}{2}c_4E$ from the closest side of SQ inside the interior of SQ. Then $3c_4E$ points of the flank must be inside SQ, but this implies that at least c_4E intended boundary points have been lost, which is impossible. Therefore, the flank completing the lines of zeros on the left must lie to the left of the line $x = -\frac{1}{2}(L - c_6\sqrt{LE}) + \frac{3}{2}c_4E$ in the vertical interval and the flank completing the lines of zeros on the right must lie to the right of the line $x = \frac{1}{2}(L - c_6\sqrt{LE}) - \frac{3}{2}c_4E$ in the vertical interval. ■

Let RT be the region bounded by the two flanks and the lines $y = \pm(\frac{1}{2}(L - c_6\sqrt{LE}) - c_7\sqrt{LE})$. We now show that greater than $L - \frac{L}{n}$ non-flank strings pass through (R_C and) RT and exit through its top and bottom sides. (We will show that no prefix or suffix bit may be contained in RT.) In addition, we show we may assume all zeros contained in the sparse substrings of the non-flank strings are straight. This is the large collection of approximately straight and parallel strings we have been searching for. An important fact to note is that the y-coordinate (vertical coordinate) of any point contained in the sparse substring of these strings is confined to lie in an interval of length $(c_6 + 2c_7)\sqrt{LE} + 1$; that is, bit number l of the internal substring must lie in the vertical interval of length $(c_6 + 2c_7)\sqrt{LE} + 1$ centered at $\frac{L}{2} - l$. Setting $c = 4(c_6 + 2c_7)$, so that the shortest number of ones between two zeros in the sparse substring is $4(c_6 + 2c_7)\lceil\sqrt{LE}\rceil$, it is clearly impossible for a zero in the sparse substring to be adjacent to a zero from the same string. Further, and very importantly, if the zero is bit number l of the internal substring of one of our collection of strings, then it may not be adjacent to any other zero in another string in the collection other than the zero (if it is a zero) in bit number l of the other string. These facts will allow us to find a Hamilton path using the collection of strings. They will also be important in the two lemmas and two corollary below where we find the collection of strings. Note, finally, that the zeros and all the code bits in the sparse substrings of the strings in the collection must lie in the vertical interval, $[-2E(c\lceil\sqrt{LE}\rceil + 1) - (c_6 + 2c_7)\sqrt{LE}, 2E(c\lceil\sqrt{LE}\rceil + 1) + (c_6 + 2c_7)\sqrt{LE}] \subseteq [-(2cE + c_6 + 2c_7))\lceil\sqrt{LE}\rceil - 2E, (2cE + c_6 + 2c_7))\lceil\sqrt{LE}\rceil + 2E]$.

Lemma 13 *Any non-flank string which passes through the region, RT', which is bounded by the two flanks and the lines $y = \pm((2cE + c_6 + 2c_7))\lceil\sqrt{LE}\rceil + 2E)$ must exit RT through its top and*

bottom sides.

Proof: We first show that no prefix or suffix bit may be contained in RT. Recall that lines of vertical zeros without associated losses and completed by flanks are guaranteed to lie on the top and bottom sides of RT or just above and below these sides. No prefix or suffix bit may lie on these lines. Further, since no prefix or suffix bit may lie in SQ, the whole prefix or suffix must be contained in the region bounded by the closest flank and side of SQ, and the two lines. However, this would cause the loss of too many intended boundary points contained in the closest flank. Thus, no prefix or suffix bit may be contained in RT (and we also note for future reference that there can be no intended corners inside RT).

Let s' be a non-flank string passing through RT'. String s' must exit RT through two points contained in its top and/or bottom sides. We, next, eliminate the possibility of s' forming a U-shape with respect to either the top or bottom side using the same argument as used in Lemma 11. We need only the assumption that $(c_6 + 2c_7)\sqrt{LE} + (4cE + 2c_6 + 4c_7)\lceil\sqrt{LE}\rceil + 4E \leq \frac{L}{10}$, which is true for $n \geq 10$. Thus s' exits RT through its top and bottom sides as desired. ■

Lemma 14 *There are at least $L - c_6\lceil\sqrt{LE}\rceil - 6c_4E$ strings which pass through (R_C and) RT and exit through its top and bottom sides.*

Proof: We prove that $L - c_6\sqrt{LE} - 6c_4E$ strings pass through the region $RT' \cap SQ$ contained in RT by again making use of horizontal lines of vertical zeros. It suffices to show that s has a sparse vertical zero without a loss in vertical distance one. For then, a horizontal line of vertical zeros must be formed and we are guaranteed by the previous lemma that all strings containing the zeros which pass through $RT' \cap SQ$ must exit RT through its top and bottom sides. Further, by the observations made before the previous lemma, the strings in $RT' \cap SQ$ containing the zeros are distinct. Finally, since no flank string may pass through SQ further than $\frac{3}{2}c_4E$ from the corresponding side of SQ (as noted in Corollary 12) to end the line, there are certainly at least $L - c_6\lceil\sqrt{LE}\rceil - 6c_4E$ strings passing through RT. (We removed an extra factor of $3c_4E$-2 because later it will be useful to assume these strings are far from the flanks.)

We verify that there is always at least one vertical zero without a loss in vertical distance one contained in the sparse substring of s. Any horizontal zero in the sparse substring must have a loss within vertical distance $(c_6 + 2c_7)\lceil\sqrt{LE}\rceil + 1$ since, if there is no loss within vertical distance $(c_6+2c_7)\lceil\sqrt{LE}\rceil$, then a vertical line of zeros extending upward of length $(c_6+2c_7)\lceil\sqrt{LE}\rceil+1$ must be formed. However, because of the large distance between zeros noted in the paragraph before the previous lemma, the next bit in the line must be a one or empty, resulting in a loss. Next, the only way for a sparse bent zero in s not to have a loss within vertical distance two (or distance 2), is if it is adjacent to two other zeros:

$$\begin{array}{cc} 1 & 1 \\ | & | \\ 0 & 0-1 \\ & 0-1. \end{array}$$

However, the two zeros which are adjacent to the bent zero, neither of which may be flank zeros, must be contained in the same string since non-flank strings (contained in RT') exit RT through

its top and bottom sides. Yet this is impossible due to the limitations on the proximity of zeros. Therefore, all sparse bent zeros must have a loss within vertical distance two (and distance two). Since losses within vertical distance $(c_6 + 2c_7)\lceil\sqrt{LE}\rceil + 1$ are uniquely attributable to the sparse zeros in a string and there are at least $E + 1$ sparse zeros, there must be a vertical sparse zero without a loss in vertical distance one. ■

Corollary 15 *There exist greater than $L - \frac{L}{n}$ non-flank strings which pass vertically through RT such that all their sparse zeros are vertical or horizontal.*

Proof: There are at most 8 points within distance 2 of a loss, so there are at most $8E$ bent zeros. Thus at least $L - c_6\lceil\sqrt{LE}\rceil - 6c_4E - 8E$ strings pass vertically through RT such that all their zeros are straight. Assuming $c_6\lceil\sqrt{LE}\rceil + 6c_4E + 8E < \frac{L}{n}$, true for $n \geq 10$, we are done. ■

Therefore, there is at least one representative of every node passing vertically through RT. Let $(s_{k_1}, s_{k_2}, \ldots, s_{k_m})$ be the left-to-right order of the collection of strings in Corollary 15. We claim that for $l \in [m-1]$, the number of losses involving strings s_{k_l} and $s_{k_{l+1}}$ and points in between must be at least the Hamming distance of the two strings. Recall the Hamming distance of the two strings is equal to the number of points where a zero bit from the Trevisan code of one node corresponds a one bit from the Trevisan code of the other node (the bits are at the same position in the Trevisan code). Suppose there is a code vertical zero in s_{k_l} which corresponds to a code one in $s_{k_{l+1}}$, and let us say s_{k_l} is to the left of $s_{k_{l+1}}$. Then there must be a line of vertical zeros extending from the code zero to the right, and this line must end before or at the time it reaches $s_{k_{l+1}}$. None of the zeros in the line can be in string $s_{k_{l+1}}$ and the loss ending the line must be contained in the window consisting of the last zero in the line and its neighbors and the three adjacent points to the right. Therefore, since the loss must involve points in the strings s_{k_l} and $s_{k_{l+1}}$ and/or points in between the strings and the loss is within vertical distance one of the code zero, the loss may be uniquely attributed to the pair of differing code bits in the two strings. Similarly, if the code zero is horizontal, using similar windows, there must be a loss involving points between the strings within vertical distance at most $(c_6 + 2c_7)\lceil\sqrt{LE}\rceil + 1$. The loss may also be uniquely attributed to the pair of differing code bits. Thus, if s_{k_l} and $s_{k_{l+1}}$ correspond to different nodes, there are are at least $7n$ losses which can be attributed to the two strings due to the Trevisan code. However, since all n nodes are represented in the collection of strings, the total number of losses contained in RT is at least $7(n-1)n = E$. Consequently, there may only be $n-1$ transitions between different nodes, and the order, $(v_{j_1}, v_{j_2}, \ldots, v_{j_n})$, of the nodes corresponding to the strings above, deleting adjacent occurences of the same node, must correspond to a Hamilton path in G.

In fact, we have found our desired Hamilton path from v_1 to v_n. Assume without loss of generality that the flank on the left side of R_C corresponds to v_1, that is, the flank is s_1. (This is just a reflection of the folding f.) We prove:

Lemma 16 *It must be true that $v_{j_1} = v_1$ and $v_{j_n} = v_n$.*

Proof: We show that $v_{j_1} = v_1$; it follows by symmetry that $v_{j_n} = v_n$. First, we note that all the sparse zeros contained in s_{k_1} must be vertical. The reason is that there are always two losses attributable to sparse horizontal zeros (one above the zero and one below), and only one loss may

be lie between the strings s_{k_1} and s_{k_2}. Since we are already guaranteed E losses which lie between the strings s_{k_1} and s_{k_m}, there can be no other losses outside this region.

Next, we show that all the strings passing through RT' between s_1 and s_{k_1} also represent v_{j_1}. It suffices to verify that there are no sparse bent zeros in any of these strings. For all the strings except for the string closest to s_1 this is easy. We simply use the same argument as was used in Lemma 14 to show that there must be a loss within distance one of a string containing a sparse bent zero. Regarding the string next to s_1, call it s'', we may rule out any sparse bent zeros facing toward the right using the same argument. In terms of any sparse bent zero which faces toward the left, there can be no intended corner within distance one of the bent zero's neighbors (this was noted in Lemma 13), and the zero may not be adjacent to two other non-flank internal zeros. Thus, using the proof of Lemma 5, there must be a loss within distance 6 of the zero; however, this is impossible since s'' is far from s_{k_1}.

Finally, we show that a sparse zero contained in s_1 which is (horizontally) adjacent to a point of s'' must lie in a vertical interval of length $(c_6 + 2c_7)\lceil\sqrt{LE}\rceil$ centered at $\frac{L}{2} - l$, if the zero is the lth internal bit contained in s_1. This fact will complete the proof, for then s_1 must have zeros in exactly the same Trevisan code positions as s''. Since the Trevisan code contained in s_1 has at most as many zeros as the code in s'', the codes must then be identical. Therefore, $v_{j_1} = v_1$.

To show that a sparse zero of s_1 which is (horizontally) adjacent to s'' must lie in the desired interval, it suffices to show that at least $L - (c_6 + 2c_7)\lfloor\sqrt{LE}\rfloor$ internal points of s_1 with distinct y-coordinates are horizontally adjacent to points of s'' and are contained in a vertical interval of length $L - (c_6 + 2c_7)\lfloor\sqrt{LE}\rfloor$. Inside RT, $L - (c_6 + 2c_7)\lfloor\sqrt{LE}\rfloor$ internal points of s'' with distinct y-coordinates face s_1 horizontally (in a vertical interval of length $L - (c_6 + 2c_7)\lfloor\sqrt{LE}\rfloor$). Again making use of Lemma 5, none of these points may be a bent zero. Since all the points must then be ones or vertical zeros and no losses within distance two of s'' are allowable, the points must all be adjacent horizontally to distinct internal points of s_1. This completes the proof. ■