Protein Folding, Spin Glass and Computational Complexity[1]

Aviezri S. Fraenkel

Abstract. A reduction from "Ground State of Spin Glass" in statistical mechanics to a minimum-energy model of protein folding is made, which shows that the latter is NP-complete (high complexity). The reduction approximates true folding of a protein. The method also enables to show that even if the backbone of the protein is fixed, the folding of the side-chains is NP-complete. In a separate second part, the possibility of synthesizing proteins to solve arbitrary instances of the spin glass problem is speculated upon.

1. Introduction

The motivation for this work is the speculation of exploiting nature's capability of protein folding to solve computationally intractable problems. One way of investigating this idea is to encode known NP-complete problems in terms of protein folding. The main content of this paper is to do this for the spin glass problem. We construct a protein that achieves the encoding, i.e., the folded protein provides a solution to spin glass. More precisely, albeit incidentally, we prove that a minimum energy model of protein folding is NP-complete.

In [**Fra1990**] the NP-completeness of a minimum energy model of protein folding was announced, and the proof appeared in [**Fra1993**]. Independent proofs were given by [**NgM1992**], [**UnM1993**] and [**HaI1996b**]. See also [**HaI1997**]. All of these proofs were rather artificial, and didn't involve true folding of a protein. In [**PaP1996**], an embedding of a chain in a grid was shown to be NP-complete, where the chain is over an unbounded alphabet, and the score is the number of adjacent matching string symbols. In the present paper we come a little closer to true protein folding, in that the number of amino acids is only three, the amino acids adjacent along the chain are also adjacent on the lattice, and a process of folding is involved which approximates that of true proteins. The problem "Ground State of a Spin Glass" which models phase transitions in statistical mechanics, magnetic fields, etc., bears a similarity to protein folding. In both, the steady state is a free energy conformation. We also mention that in [**HaI1996a**], [**HaI1996b**] and [**HaI$\geq$1997**] good approximations to optimal folding in a minimal energy model were given.

1991 *Mathematics Subject Classification.* Primary 92-08; Secondary 92D20

[1]Expanded form of invited paper presented at Third Annual DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23-25, 1997.

We chose the spin glass problem, whose general case is also NP-complete (see [**Joh1983**]), to reduce from to protein folding, in the hope of encoding an NP-complete problem more realistically in terms of protein folding. This is of importance if we hope to use protein folding to solve computational problems, especially NP-complete problems, very fast. Of course the present reduction provides, in addition, another proof of the NP-completeness of protein folding. The protein we generate is still rather restricted in its allowed movements. All rotations are restricted to be in a fixed plane, but some rotations are at least unrestricted in the angle of rotation, subject only to the lattice restriction, i.e., the vertices have to be at integer coordinates. This situation is actually somewhat similar to that of real proteins. See e.g., [**NMK1994**] §1: "Although the bond lengths and angles in a protein can be predicted easily since they cannot vary much, torsion angles (rotations about bonds) are not easily predicted. Rotatable torsions tend to have three preferred values, none of which can be ruled out a priori." A similar statement appears in [**Ree1988**]. The angle restrictions in real proteins stem from spatial constraints, since folded proteins are densely packed, so distinct atoms cannot occupy the same location.

We now begin at the beginning.

A protein is a sequence of amino acids, created as an essentially linear sequence from the well understood genetic code inherent in a DNA chain, with semi-rigid bonds (peptide bonds) between pairs of amino acids which are adjacent in the linear sequence, and other, less rigid bonds, between some other amino acids. The protein then folds into a complex 3-dimensional *native conformation*, which determines its biological functions, such as the geometry of receptor sites, enzymatic properties, etc. The folding mechanism is not known, but it is believed that the native folded conformation of a protein is its lowest free energy state [**Anf1973**]. Typically a protein consists of 1000–20,000 atoms, and has a "diameter" of 35–100Å (1Å=10^{-8} cm).

One approach to model protein folding is to consider the protein to be a collection of hard impenetrable spheres (atoms) held together by elastic strings (covalent bonds). The atoms have electric charges that obey Coulomb's law [**LeL1969**]. The electromagnetic force between any two atoms diminishes as d^{-2}, and the potential energy as d^{-1}, where d is the distance between a pair of interacting atoms. The interaction is really between every atom of the protein and every atom in the universe! Yet within about one second the protein attains its final native 3-dimensional conformation. Some chemical physicists simulate this system, where they will typically neglect forces between atoms at distance >6Å. Under this simplifying assumption, simulation of one nanosecond (10^{-9} sec) of the protein folding process still takes some 150 hours on a modern main-frame computer [**LeS1988**]. For more information about protein folding see e.g., [**GiK1990**] and [**NMK1994**].

In §2 we give some background on computational complexity and NP-completeness, and in §3 we present a 3-dimensional model of protein folding and prove it to be NP-complete. In the final §4 we speculate about possible ramifications of this and other NP-complete models of fragments of nature, leading to a brief discussion of how this proof, or rather modifications thereof, might lead to a fast computation of the spin glass problem.

2. Computational Complexity and NP-Completeness

The computational complexity of a problem is usually measured in terms of the number of "steps", i.e., the "time" required to solve it, as a function of the problem's input size. A problem is called *tractable* if this function is polynomial; *intractable* otherwise. Thus sorting n integers is tractable: it can be done in $O(n \log n)$ comparison steps. The following reasons motivate this convention:

1. Normally, only tractable problems can be solved on a computer in reasonable time. Suppose that each of the problems π_1, π_2, π_3 has input size n and that the best algorithms (= "lower bounds") for solving them need n, n^2 and 2^n steps respectively. If the rate of our machine is 10^6 steps/second, then for $n = 60$, π_1 requires 0.00006 sec for execution, π_2 0.0036 sec and π_3 366 centuries!

2. The input size of a problem that can be solved in a reasonable fixed time is of practical value only for tractable problems. Thus for, say, five hours of uninterrupted computation, π_1 can have input size 18×10^9; π_2 has size 13.4×10^4 and π_3 only 34. Moreover, any 10-fold gain in speed that technological advances may yield increases the size of a problem with an $O(n^k)$ algorithm that may be solved in a fixed time by the *factor* $10^{1/k}$, whereas the size of a problem with an $O(c^n)$ algorithm is increased only by an *additive* amount of $\log_c 10$ ($c > 1$ a constant).

3. The most simplistic approach to solving a problem is to explore its entire "search tree", i.e., searching through all possibilities. Except for trivial problems, this search constitutes an exponential algorithm. Thus a problem whose best algorithm is exponential has often no essentially better algorithm than to search through all or most possibilities.

(In reality the world is not so simple; mostly in the pessimistic direction: there are problems which are polynomial and still appear to be intuitively intractable in two different senses! This is implied by the recent Robertson-Seymour theory in graph minors. See e.g., [**RoS1996**].)

Many problems can be shown to be tractable, simply by producing a polynomial algorithm for them. Very few problems can be proved to be intractable at present, however. For the bulk of interesting problems, both tractability and intractability appear to be rather difficult to establish at this time. For a large subset of them we can do a next best thing, which is to establish *completeness*, such as NP-completeness. For the moment we restrict attention to *decision problems*, i.e., problems for which the answer is YES or NO.

A decision problem π is *NP-complete* if:

(i) Given any solution for π, its validity can be *verified* in polynomial time (but there may not be a deterministic polynomial algorithm for *finding* a solution; we say that the problem has a "nondeterministic" algorithm). So the P of NP stands for Polynomial, N for Nondeterministic.

(ii) If π can be shown to be tractable, then all NP-complete problems are tractable; if π can be shown to be intractable, then all NP-complete problems are intractable.

Since the best known algorithms for any NP-complete problem are at present non-polynomial, all NP-complete problems are presently "practically intractable", or "conditionally intractable". For a thorough treatment of NP-completeness see [**GaJ1979**].

A common way to prove that a problem π is NP-complete consists of three phases:

(a) *NP-Membership*. Show (i) directly. This is usually, but not always, the easy part of the proof.

(b) *Construction*. Select an NP-complete problem π', consider an arbitrary generic *instance* x of π' and select a function f such that $f(x)$ is some instance of π and $f(x)$ is constructed in time which is polynomial in the size of x. This phase is normally called the polynomial *construction*.

(c) *YES-Equivalence*. Show that the answer to x is YES if and only if the answer to $f(x)$ is YES.

Phases (b) and (c) together are called a *reduction* of π' to π. Notation: $\pi' \propto \pi$.

The intuitive meaning of (a) and (b) should be clear: given an arbitrary instance x of π', it is transformed polynomially into a particular instance $f(x)$ of π (Fig. 0). If π is polynomial, then, in particular, $f(x)$ can be decided in polynomial time; and the answer to $f(x)$ is YES if and only if the answer to x is YES. Thus the polynomial algorithm for solving π and the polynomial f constitute a polynomial algorithm for solving π'. Equivalently, π' intractable implies π intractable.

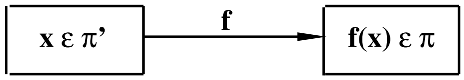

FIGURE 0. THE INTUITIVE MEANING OF A REDUCTION.

Within the so-called *Turing machine* model (see e.g. [**GaJ1979**]), it is customary to denote by P the set of all tractable problems, and by NP the set of all problems whose solutions can be verified in polynomial time. Then clearly $P \subseteq NP$. A major unsolved problem in Theoretical Computer Science is whether $P = NP$ or not. It is customarily conjectured that $P \neq NP$. Any NP-complete problem belongs to the hardest problems in NP, in the sense that if $P \neq NP$, then the NP-complete problems are intractable.

Consider the problem

GROUND STATE OF A SPIN GLASS (GSG). Given positive integers M and N, the 3-dimensional grid graph $\Gamma = (T, E)$, where the set T of vertices consists of the integer-coordinate points (x, y, z) satisfying $1 \leq x \leq M$, $1 \leq y \leq N$, $1 \leq z \leq 2$, and whose edges connect each pair of vertices that are adjacent in one of the three directions, an *interaction weight* $J(e)$ with values in $\{-1, 0, 1\}$ for each edge $e \in E$, and a negative integer H. Is there an assignment of a *spin* $s(u) \in \{-1, 1\}$ for all $u \in T$, such that the "ground state spin energy" is at most H, i.e., such that

$$\sum_{(u,v)\in E} J(u,v)s(u)s(v) \leq H? \tag{1}$$

This problem is known to be NP-complete; see [**Joh1983**].

3. NP-Completeness of Protein Folding

We consider the following 2-dimensional model for protein folding.

MINIMUM FREE ENERGY CONFORMATION OF PROTEIN (MEP). Given a graph $G = (V, A)$ (with vertex set V and edge set A), $V \subset \mathbb{Z} \times \mathbb{Z}$ (V consists of lattice points in the plane), $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, $A = A_1 \cup A_2$, $A_1 \cap A_2 = \emptyset$, a function $C\colon V \to \{-1, 0, 1\}$ (the charges), $K \in \mathbb{Z}^-$ (energy bound) and $L \in \mathbb{Z}^+$ (maximum distance). Specified subgraphs may rotate about vertices of V_2 (but not V_1) in restricted specified ways. Is there a rearrangement V' of V with $V' \subset \mathbb{Z} \times \mathbb{Z}$ which preserves $d(u, v)$ for all $(u, v) \in A_1$, such that the total *free energy* is at most K, i.e., such that

$$\sum_{(u,v)\in A} C(u)C(v)/d(u,v) \leq K, \tag{2}$$

and $d \leq L$, where d is the discretized Euclidean distance $\left\lceil \left((x_2 - x_1)^2 + (y_2 - y_1)^2\right)^{1/2} \right\rceil$?

We also consider the following restricted protein folding problem.

SIDE CHAINS OF PROTEINS (SCOP). Given a graph $G = (V, A)$, $V \subset \mathbb{Z} \times \mathbb{Z}$, $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, $A = A_1 \cup A_2$, $A_1 \cap A_2 = \emptyset$, a function $C\colon V \to \{-1, 0, 1\}$ (the charges), $K \in \mathbb{Z}^-$ (energy bound) and $L \in \mathbb{Z}^+$ (maximum distance), where V_2 is restricted to the middle vertices of subgraphs called "side-chains", permitting rotation by 180° into one of two stable states in a plane parallel to $x = 0$. No rotation is permitted for V_1 (the induced subgraph of which is the "backbone"). Is there a rearrangement V' of V with $V' \subset \mathbb{Z} \times \mathbb{Z}$ which preserves $d(u, v)$ for all $(u, v) \in A_1$, such that the total *free energy* is at most K, i.e., such that $\sum_{(u,v)\in A} C(u)C(v)/d(u, v) \leq K$, and $d \leq L$, where d is the discretized Euclidean distance $\lceil ((x_2 - x_1)^2 + (y_2 - y_1)^2)^{1/2} \rceil$?

Remarks.

(i) For both MEP and SCOP we will chose $L = 1$, so only atoms at the minimal distance of 1 interact. This is actually a better approximation for the interaction between amino acids, where the interaction diminishes as the distance to the 6-th power at least (see e.g., [**DDL1994**]), than for atoms. Thus MEP and SCOP can and will be considered to be models for protein folding with 3 amino acids.

(ii) The subset A_1 models peptide bonds, which are rigid, whereas A_2 resembles the other, less rigid bonds. Similarly, the subsets V_1 and V_2 model amino acids which do not permit and permit rotation, respectively.

Our purpose is to show that the decision problem MEP is NP-complete. The corresponding optimization problem — where we ask for $\min E$ rather than only $E \leq K$ — is then clearly not any easier than MEP. In general, if a decision problem π_1 is NP-complete, then its corresponding optimization problem π_2 is said to be NP-*hard*. We note that NP-hard problems are, generally, not any easier to solve than the NP-complete problems they correspond to. We also show that even the simpler problem SCOP is NP-complete.

The NP-membership of MEP or SCOP follows from the observation that, given any solution to MEP or SCOP, we have only to check that $d(u, v)$ is preserved for all $(u, v) \in A_1$ and that $E \leq K$, both of which can be done in time which is a polynomial in the input size of G.

We show GSG $\propto$ MEP. Before doing so, it is convenient to prove the following auxiliary result.

Lemma 1. *Let $x_1, \ldots, x_k$ be any real numbers, and $(\pi(x_1), \ldots, \pi(x_k))$ any permutation of $(x_1, \ldots, x_k)$. Then $\sum_{i=1}^k x_i \pi(x_i) \leq \sum_{i=1}^k x_i^2$.*

Proof. Consider the two k-dimensional vectors $\mathbb{V} = (x_1, \ldots, x_k)$ and $\mathbb{V}_\pi = (\pi(x_1), \ldots, \pi(x_k))$. As usual, the inner product of vectors $\mathbb{V}$ and $\mathbb{V}' = (y_1 \ldots, y_k)$ is defined by $(\mathbb{V}, \mathbb{V}') = \sum_{i=1}^k x_i y_i$. The well-known Cauchy-Schwarz inequality asserts that $(\mathbb{V}, \mathbb{V}') \leq \| \mathbb{V} \| \| \mathbb{V}' \|$, where $\| \mathbb{V} \| = (\mathbb{V}, \mathbb{V})^{1/2}$ is the *norm* or *length* of $\mathbb{V}$. Since clearly $\| \mathbb{V} \| = \| \mathbb{V}_\pi \|$, we have, $(\mathbb{V}, \mathbb{V}_\pi) \leq \| \mathbb{V} \|^2 = (\mathbb{V}, \mathbb{V})$, which is the desired result. ∎

Theorem 1. *The problem* MEP *is* NP-*complete.*

Proof. Given an arbitrary instance $\Gamma = (T, E)$ of GSG, we construct an instance $G = (V, A)$ of MEP. Put $L = 1$. In all our figures we denote by $\bullet$ a vertex of V with charge $+1$, by $\circ$ a vertex with charge -1, by $\times$ a vertex with charge 0, and by $\star$ a "rotation axis", i.e., a vertex in V_2, about which an attached subgraph has restricted rotation capability, as specified for each figure with rotation axes. Every rotation axis has charge 0. Informally, the expected native (folded) conformation of G will be a 3-dimensional box, as is Γ, but each vertex of Γ will transform into a small subgraph in G, which will constitute the "side chains".

Our basic building blocks are *strands*. A typical strand has a central vertex called *center*, from which emanate rays of length 1 or 2 in each of the three directions, possibly in both the negative and positive orientation for the x- and y-axes, but in only one orientation for z. Thus a center has degree 3, 4 or 5, where the *degree* of a vertex is the number of edges impinging on it. Edges joining vertices in the figures all belong to A_1; they have length 1 unless otherwise specified. Vertices u, v at distance 1 of each other with no edge drawn between them form an edge $(u, v) \in E_2$. In a *charged* strand, the center has charge -1 or 1, and adjacent vertices have alternating charges; and in a *neutral* strand, the center and other vertices have charge 0, except that the *ends* (vertices of degree 1), all have degree -1 or 0; or all have degree 1 or 0 (Fig. 1). The centers of the strands simulate the location and degrees of the vertices of the original instance of Γ in a sense to become clear in the sequel.

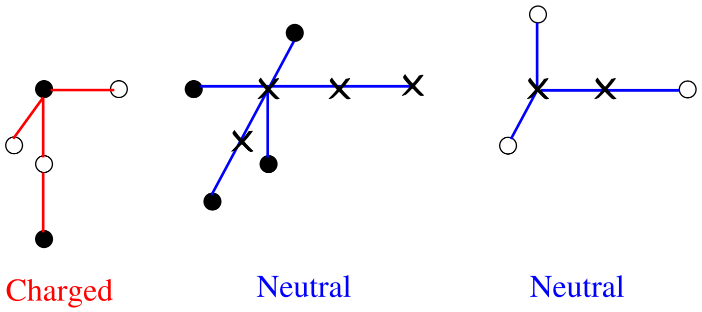

FIGURE 1. EXAMPLES OF STRANDS.

For a charged strand S with vertical ray pointing in the negative z-direction, there is a "mirror-image" called *complement* strand $\overline{S}$ two units "above" S, and parallel to it in the y-axis, antiparallel in the x- and z-axes. If the vertical ray of S points in the positive z-direction, then the complement, analogous to the above, is two units "below" S (Fig. 2). In either case, the orientations of the rays in the x- and z-axes are reversed in the complement, and all the corresponding charges are *complemented*, where 1 and -1 are complements, and the complement of 0 is 0. The two strands are joined via rotation vertices (with charge 0) in the vertical plane between S and $\overline{S}$. The joining network, together with S and $\overline{S}$ constitute a *frame*.

Analogously to the frames there are the *shells*: Four units above or below a neutral strand S there is its complement $\overline{S}$, parallel in the y-axis, antiparallel in the x- and z-axes, and complemented charges (Fig. 3). The two middle vertices of the joining network are rotation vertices, and the four others are vertices with charge 0 which cannot rotate about themselves. A frame or shell is also referred to as a *side-chain*.

It's time to explore the global aspects of the construction. We illustrate with an instance of GSG with $M = 3$, $N = 4$, complete with its J-weights, as depicted in Fig. 4. It will aid us in shaping

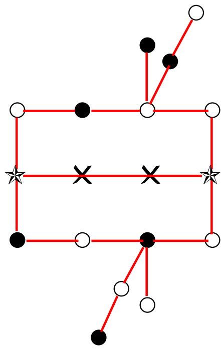

FIGURE 2. EXAMPLE OF A FRAME.

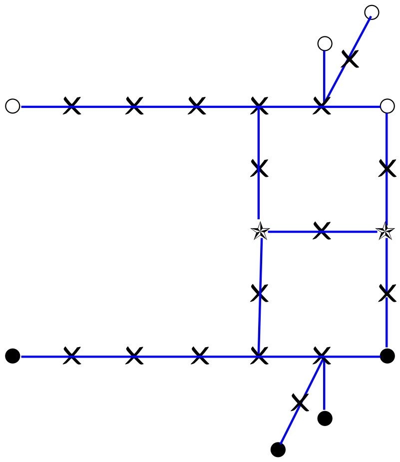

FIGURE 3. A SHELL.

our view of the overall construction. An important feature of Γ is that it is *bipartite*, i.e., its vertex set T can be partitioned into two disjoint subsets T_0 and T_1 such that all edges of Γ join vertices between T_0 and T_1. This follows from the fact that all cycles in Γ are evidently even. Without loss of generality we may assume that the "origin" (vertex at (1,1,1)) is in T_0.

To fix the construction, view each shell as simulating a vertex in T_0, and each frame as simulating a vertex in T_1. Frames and shells are, in fact, adjacent. There are three types of bonds for tying together shells and frames. The first involves *glue*, to be defined below. For the moment it suffices to say that the twin *strings* with oppositely charged vertices facing each other at distance 1, on top of Fig. 5, constitute glue. At their bottoms, the two strings impinge on rotation vertices in V_2. The rotation of each string can be by any amount in a plane which, in Fig. 5, is parallel to $y = 0$. Only in the position shown in Fig. 5 the charges are at distance 1, and the strings "stabilize". Notice that the glue is in a plane parallel to the plane of the rest of the figure, 1 unit "behind" it.

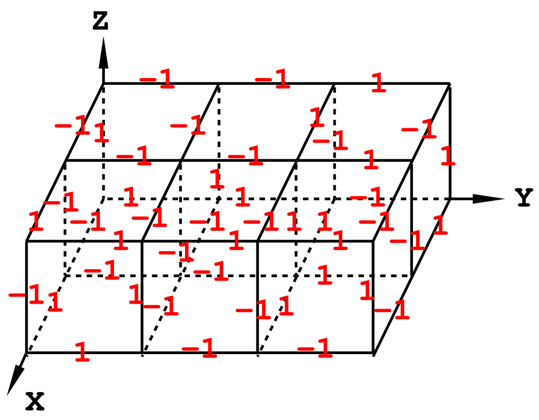

FIGURE 4. INSTANCE OF GSG WITH M=3, N=4.

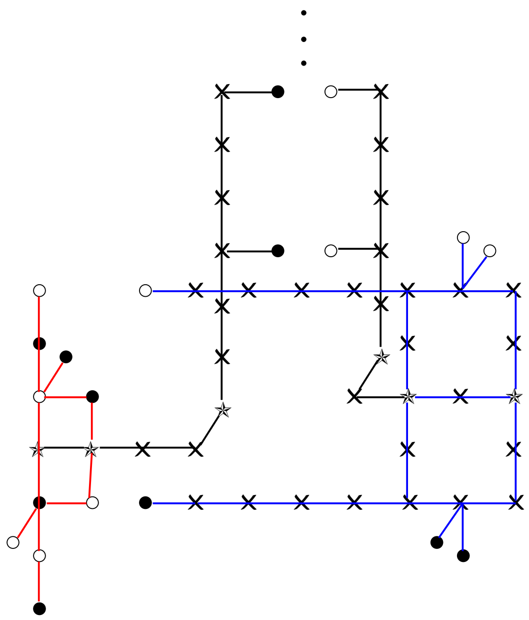

FIGURE 5. INTERCONNECTION OF FRAMES AND SHELLS: I. GLUE

The second type of bond (Fig. 6) is by means of the *backbone*, which impinges on every frame and shell (as seen in Fig. 8). The third bond is *implicit* (Fig. 7). It looks as if there is no bond at all, but the frame and shell are held together by the backbone, which intersects them in a different plane.

Whatever the bonding of a shell and frame is, each can rotate, independently of each other, by 180° about two rotation axes (Figs. 5, 6, 7); so the two stable states for a side-chain are both inside

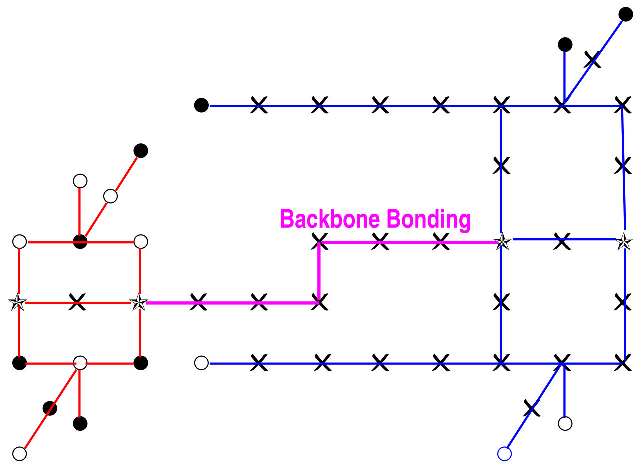

FIGURE 6. INTERCONNECTION OF FRAMES AND SHELLS: II. BACKBONE

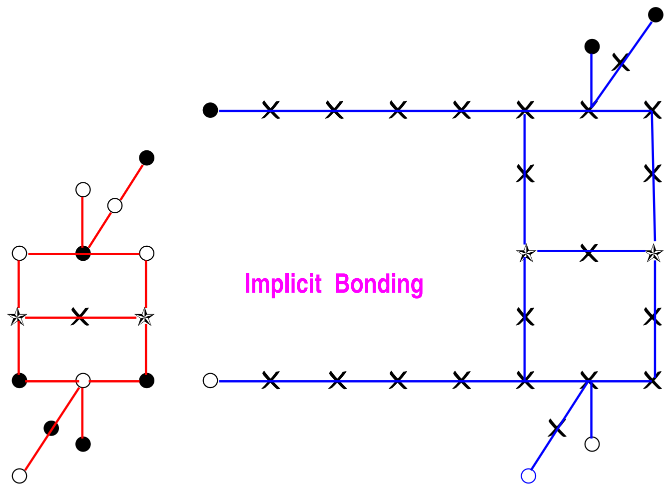

FIGURE 7. INTERCONNECTION OF FRAMES AND SHELLS: III. IMPLICIT

a plane parallel to $x = 0$, with either S above and $\overline{S}$ below, or the other way around. The aggregate of the joining edges, together with the *glue*, to be specified below, constitutes the *main chain*, or simply the *chain*. Note that there are MN side-chains intersecting the plane $z = 1$, and that many intersecting the plane $z = 2$.

The chain is expected to fold into the native conformation, portrayed schematically in Fig. 8, which is also known as the *backbone*. The chain giving rise to the backbone is depicted in Fig. 9. It is the glue which is likely to cause the chain to fold into the backbone.

The glue g_i ($i \in \{1, \ldots, 2M\}$) consists of a run of $i + 7$ negatively charged vertices spaced at distances 3, and another run of $i + 7$ positively charged vertices spaced at distances 3, arranged on

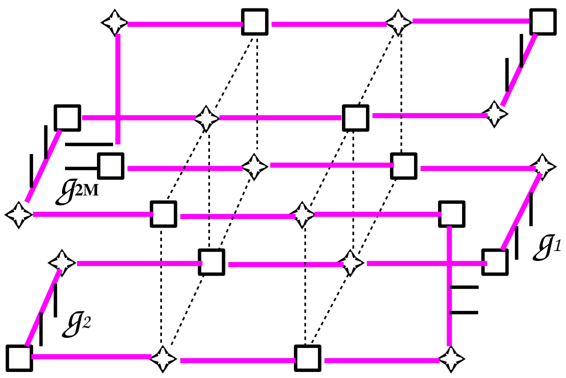

FIGURE 8. SCHEMA OF EXPECTED BACKBONE. OVALS DENOTE FRAMES, SQUARES— SHELLS. THE iTH GLUE IS g_i. (THE DOTTED LINES ARE NOT PART OF THE BACKBONE.)

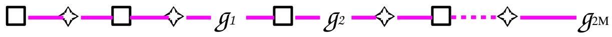

FIGURE 9. SCHEMA OF CHAIN: FRAMES (OVALS), SHELLS (SQUARES) AND GLUE (g).

edges of adjacent vertices with charge 0, which are called strings. Each vertex of a glue is encased on 5 sides with vertices of charge 0 at distance 1, and only the side expected to interact with the matching vertex of opposite charge is unblocked. (In Fig. 5, the encasing of the glue is not shown, to avoid cluttering.) Glue strings with positive charges impinge on a subset of frames, and strings with negative charges—on a subset of shells. The rotation vertices of the glue permit rotation of the glue strings by any angle, but in one plane only. The unblocked interaction direction of the oppositely charged vertices of g_i is designed to make the expected folding into the backbone (Fig. 8) possible. Fig. 10 portrays two vertices of g_i, The expected folding of this glue is shown in Fig. 11.

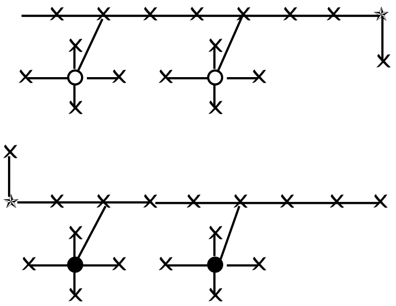

FIGURE 10. TWO VERTICES OF GLUE OF POSITIVE AND NEGATIVE CHARGE.

The glue bonds $2M$ shells and frames, as shown in Fig. 8, which also shows that in the expected folding, the beginning and end of the chain attach to each other, by virtue of g_{2m}.

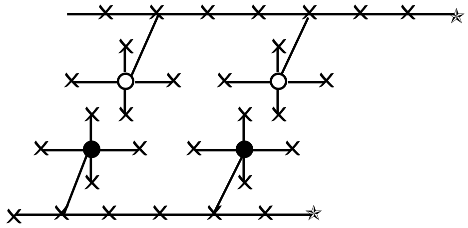

FIGURE 11. EXPECTED FOLDED FORM OF GLUE OF PREVIOUS FIGURE.

In Fig. 12 we see the side chains in their expected native conformation. Only one complete frame and one "compressed" shell with a small part of the chain hugging them is shown, to avoid cluttering the figure.

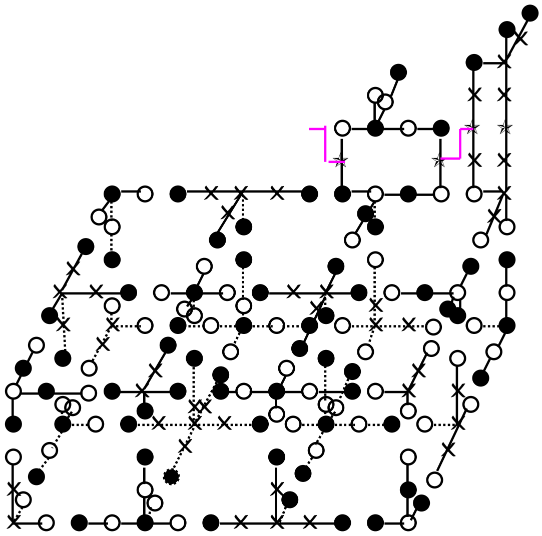

FIGURE 12. ESSENTIAL PART OF THE EXPECTED NATIVE CONFORMATION.

At edges of Γ where $J = -1$ or $J = 0$, the strands of the frames have length 1, and at edges where $J = 1$, the strands of the frames have length 2. The situation for the strands of the shells is in a way reversed: they have length 6 for $J = -1$ or $J = 0$, and length 5 for $J = 1$. (The larger length of the shell-strands enables to accommodate the glue—see Fig. 5.) This is illustrated in Fig. 12 for the values of J given for the instance of GSG of Fig. 4, except that the shell-strands are shortened to length 3 or 4, to enable portraying the essential features of the construction in one figure. Note that in the expected native conformation, each edge of the emerging grid G has an end-vertex of a frame-ray at distance 1 from the end-vertex of a shell-ray: they *dock* with each other, forming an edge in E_2. The two docking rays constitute the *docking site*. The two end vertices at a docking site are the *docking vertices*.

The end vertex of a shell-ray has charge 0 if and only if it corresponds to an edge of Γ with $J = 0$ (for S and $\overline{S}$). All other end vertices of a shell-ray have charge 1 in S, and -1 in $\overline{S}$. The construction is completed by putting $K = H + 2m + 2m_1 + M(2M + 15)$, where $m = |E|$ and m_1 denotes the number of edges of Γ with weight $J = 1$.

Any instance of Γ has 8 vertices of degree 3, $4((M-2)+(N-2))$ vertices of degree 4 and $2(M-2)(N-2)$ vertices of degree 5. Thus,

$$m = 3 \times 4 + 8(M + N - 4) + 5(M - 2)(N - 2) = 5MN - 2(M + N),$$

which is 46 for the instance of GSG depicted in Fig. 4. Since $m_1 \leq m$, the construction is polynomial.

Suppose now that the given instance of GSG has a spin assignment $s(u) \in \{-1, 1\}$ for all $u \in T$, such that (1) holds. We show that there exists a folding of G satisfying (2).

The charged strands, of all the frames, contribute $-2(m_{-1} + m_0) - 4m_1 = -2m - 2m_1$ (-67 for Fig. 12), where $m_{-1} + m_0 = m - m_1$, and m_{-1} is the number of edges of Γ with weight $J = -1$; m_0 the number of edges with $J = 0$. We claim that the folding of G such that (2) holds is the expected one, where the positive and negative charges of g_i match each other precisely for all i, as per Fig. 11. The total contribution of the glue is then $-\sum_{i=1}^{2M}(i + 7) = -M(2M + 15)$. It thus suffices to show that there exists a folding of the side-chains such that the docking vertices contribute energy $\leq H$, i.e.,

$$\sum_{(w,z)\in D} C(w)C(z)/d(w, z) \leq H, \tag{3}$$

where $D \subset A$ is the subset of docking vertices.

Let (u, v) be any edge of Γ. Without loss of generality we may assume $u \in T_0$, $v \in T_1$. If $s(u) = -1$, rotate the shell which simulates u such that its end vertices with charge -1 or 0 are at the docking site. If $s(u) = 1$, rotate it such that its end vertices with charge 1 or 0 are the docking vertices. If $s(v) = -1$, rotate the frame which simulates v such that its center vertex r has charge -1, and if $s(v) = 1$, rotate it such that r has charge 1. Thus $C(r) = s(v)$.

Denote by w, z, the two docking vertices in G which simulate the edge $(u, v) \in \Gamma$, where w is at the end of a frame-ray with center r simulating v, and z is at the end of a shell-ray simulating u. Note that the specified folding of the shell implies that $C(z) = s(u)$ if $J(u, v) \neq 0$; and $C(z) = 0$ if $J(u, v) = 0$. The folding of the frame shows that $C(w) = C(r) = s(v)$ if $J(u, v) = 1$, because then the frame-ray has length 2; and $C(w) = -C(r) = -s(v)$ if $J(u, v) \in \{-1, 0\}$. This information is summarized in Table 1. Since its middle-column and last column turn out to have identical entries, we conclude that $J(u, v)s(u)s(v) = C(w)C(z)/d(w, z)$ for all cases. It follows that (1) implies (3).

Conversely, suppose that there exists a folding of G which satisfies (2). We have to show that there exists a spin assignment $s(u) \in \{-1, 1\}$ for all $u \in T$ satisfying (1). Since the frames contribute $-2m - 2m_1$ to K, we have actually

$$\sum_{(w,z)\in P} C(w)C(z)/d(w, z) \leq H + M(2M + 15),$$

where $P = D \cup Q$ and Q is the set of edges each of which intersects the glue. Thus a vertex u of the glue which folds to within distance 1 from a vertex v of a side-chain contributes an edge $(u, v) \in Q$.

TABLE 1.

$J(u,v)$	$s(u)$	$s(v)$	$Js(u)s(v)$	$C(w)$	$C(z)$	$C(w)C(z)/d(w,z)$
1	-1	-1	1	-1	-1	1
1	-1	1	-1	1	-1	-1
1	1	-1	-1	-1	1	-1
1	1	1	1	1	1	1
-1	-1	-1	-1	1	-1	-1
-1	-1	1	1	-1	-1	1
-1	1	-1	1	1	1	1
-1	1	1	-1	-1	1	-1
0	-1	-1	0	1	0	0
0	-1	1	0	-1	0	0
0	1	-1	0	1	0	0
0	1	1	0	-1	0	0

Consider first the case where the glue forms a closed set, i.e., glue vertices are adjacent to each other only. Since glue vertices are blocked by 0-charges on 5 sides, at most one pair of vertices of glue can interact with each other. Putting $x_i = i + 7$ in Lemma 1, we see that the smallest energy contribution is achieved when all the glues are matched up perfectly; in other words, any glue association contributes $\geq -M(2M + 15)$. Thus (3) holds a fortiori for any closed glue folding.

Secondly, consider the case where the glue interacts with the side-chains. Note that the rotation vertices keep the glue strings and the strands in directions parallel to the $x-$, $y-$ and $z-$axes only. The smallest glue run has length 8, and from Fig. 12 we see that the side-chains contain at most 7 consecutive vertices at distance 1 with charge $\neq 0$. Note further that each glue-vertex is encased by 0-charges at distance 1 from it from 5 sides. Thus no clusters of nonzero charges can be formed. It follows that any interaction between glue and side-chains has a larger contribution than the perfect glue match-up, i.e., we see that the contribution for this case is also $\geq -M(2M + 15)$. Thus (3) holds in either case.

So it turns out, incidentally, that the folding may possibly *not* be the expected one, which is, in a way, analogous to what seems to happen in the prion protein, where α-helices are transformed into β-sheets, precipitating a diseased folded state, which is claimed to induce the mad-cow syndrome [**HBDPC**]. In any case (3) holds, and this suffices to assign spins in $\{-1, 1\}$ to the vertices of Γ to satisfy (1), as is shown now.

We use again the notation $(u, v) \in E$ with $u \in T_0$, $v \in T_1$; and z is at the end of a shell-ray simulating u; w at the end of a frame-ray with center r which simulates v. If $C(z) = -1$, then put $s(u) = -1$, and if $C(z) = 1$, put $s(u) = 1$. If $C(z) = 0$, examine the charges at the other ends of the same strand z belongs to. If any of its charges is 1, put $s(u) = 1$; if one of them is -1, put $s(u) = -1$; if all of them are 0, $s(u)$ can be assigned either -1 or 1. Further, put $s(v) = C(w)$ if w is at distance 2 from r (then $J(u, v) = 1$); and $s(v) = -C(w)$ if w is at distance 1 from r (then $J(u, v) \in \{-1, 0\}$). It then follows from Table 1 that $J(u, v)s(u)s(v) = C(w)C(z)$ in all cases, so (3) implies (1). ■

Corollary 1. SCOP *is* NP-*complete.*

Proof. We use a small modification of the proof of Theorem 1, namely we replace the glue with its vertices in V_2 by backbone bonding as in Fig. 6, and put $K = H + 2m + 2m_1$. Everything else remains the same. ∎

Remark The protein constructed in the proof of Theorem 1 can be modified in many ways, such as in the z-axis more than two layers can be used. We used only two in order to simplify the construction.

4. Speculations about Implications

Analogously to [**Fra1990**] and [**Fra1993**], we now indulge in some unscientific speculations about nature. In the present context it is aimed at the possible design of experiments for synthesizing proteins for fast computing. We first present two opposing views:

A. Nature can solve NP-complete problems in polynomial time.

The following is but a small fraction of the supporting evidence.

Each amino acid in a protein can adopt, on the average, 8 different conformations [**Pri1979**]. A relatively small protein, consisting of 100 amino acids, can thus potentially assume 8^{100} conformations. Yet nature attains the native conformation in about one second. (Note that the claim that nature assumes the global minimum free energy conformation in one second is *not* equivalent to saying that it explores all the 8^{100} potential conformations in one second: when we sort 100 distinct integers in ascending order, we don't have to test all the $100! > 100^{100}$ different permutations!)

The double helices of DNA become knotted and linked in the course of biological processes such as replication. There are methods for realizing given knots and links on DNA chains [**WaC1986**]. Moreover, the untying mechanisms follow topological transformations, so we can tell at the end of the process whether the unknot is a knot or not. The complexity of this question is a major unsolved problem in mathematics; see e.g.[**Wel1993**]. In statistical mechanics, many models have been studied to explain phase transitions [**Bax1982**]. Among them is the spin glass model GSG which we used in the proof of Theorem 1. This problem was shown to be NP-complete in [**Bar1982**]. Special cases are polynomial; see [**BMRU1980**], [**Bar1982**], [**BMRU1982**]. Nature usually manages to accomplish the phase transitions without a hitch and very fast.

Deutsch [**Deu1985**], [**Deu1989**] argues that "quantum computers" (computers based on quantum physics theory) can be constructed that can carry out in polynomial time computations which require exponential time. Bennett and Brassard [**BeB1989**] proposed and implemented practical protocols —realizing an idea of Wiesner [**Wie1983**] — and constructed a quantum counter for a certain public key cryptography application. See also [**Bra1988**], [**BrC1990**] and [**BBBSS1992**] [**BBCS1992**].

In the cryptography quantum computer, polarized photons are used to transmit digital information. Single photons (with high probability) are transmitted over a communication channel in one of 4 polarizations. There is no complexity gain in this device, yet it makes a dent in the Turing machine model: a Turing machine M can, by eavesdropping, learn the secrets being exchanged by two interactive Turing machines M_1 and M_2, without M_1 and M_2 learning that M has learned their secrets. But no machine can eavesdrop on two interactive quantum machines without being detected with high probability, assuming Heisenberg's uncertainty principle, which implies that eavesdropping to single photon transmission is tantamount to tampering with it.

Because of reasons of this type, it was suggested in [**Fra1990**], to try and reverse our usual scientific endeavor: in *addition* to modeling nature, studying the models, solving them or proving them to be NP-complete, etc., try to use devices of nature, such as proteins, quantum apparatus, DNA-chains etc. as black boxes, to which we input instances of NP-complete problems and output solutions in reasonable time. One of the problems to be faced here is the design of efficient input/output interfaces to proteins and to other devices of nature. This is a problem even for the large DNA chains (containing some 10^9 nucleotides).

The opposite view is:

B. Nature functions within the Turing machine model.

How then can nature's apparent capability of solving NP-complete problems be explained without being forced to conclude $P = NP$? There are various answers to this question.

(i) Nature does not necessarily achieve *global* optimization. If only a local energy minimum is achieved, say in protein folding, even if it is at bounded distance from the global minimum, it may not produce a near-optimal solution for an NP-complete problem encoded in terms of protein folding: NP-complete problems are equivalent and can be encoded in terms of each other; yet some of them have good approximation properties, and some don't; see e.g., [**ArL1997**].

(ii) NP-completeness is an asymptotic property, whereas the universe seems to be finite. Moreover, it seems that once a protein is sufficiently large, it is subdivided by nature into units of smaller size, say up to 200 amino acids per unit, which fold independently! See e.g. [**Pri1982**] and [**JaW1983**].

(iii) A problem π is polynomial if it is *universally* polynomial, i.e., if *all* its instances can be solved in polynomial time; it is NP-complete if *some* of its instances are NP-complete, though some of them may be solvable in polynomial time. In fact, NP-completeness reflects worst case behavior, but the average case behavior, under the assumption of some probability distribution, may be polynomial. Processes of nature are not necessarily universal. Thus perhaps the natural selection of nature may help to preserve proteins with polynomial folding mechanisms, and reject the others. Indeed, experiments show that some synthesized proteins may fail to fold into a stable conformation. Similarly, perhaps nature creates only very few knot types in DNA helices.

(iv) The protein folding mechanism may be encoded in the protein's amino acid sequence, analogously to the genetic code of a DNA chain, but the code is still unknown. If this is the case, then folding is not a search process, and there is no issue of complexity.

It may be difficult to decide between the views A and B at present. Some discussions on this point have been made. See e.g., [**Tra1996**]. But perhaps we can design experiments to create artificial proteins for exploiting fast computations even if the second view will turn out to be the valid one. For recent progress in protein design see e.g., [**BBD1995**], [**CDS1996**]. We don't understand the folding of proteins. Ken Dill put it thus: "Protein folding is a problem for us, but not for the proteins" (as quoted in [**Joh1997**]). However, mankind has exploited fire, dams, electricity, wireless communication, even before understanding the chemistry of fire, the hydraulics of water streams, the theory of electricity and waves. There is the possibility, if highly speculative, that we may be able to exploit protein folding for fast computing, even though we don't quite understand how nature goes about folding proteins.

It seems that the protein constructed in the proof of Theorem 1 shares a few properties of real proteins. It can fold, if unlike the folding of true proteins. The subsets A_1 and A_2 of edges resemble peptide bonds and other, less strong couplings respectively. The subset V_2 of vertices enables restricted rotations of specified subgraphs. The glue is possibly analogous to the subset of five generic nonpolar amino acids, any of which can be placed at strategic locations on the chain, which produce a similar pattern to an entire family of synthetic and natural proteins [**KSXBH1993**], [**WeH1995**].

For the purpose of illustration, to solve an arbitrary instance of GSG with given J-values on the edges, using the unrealistic protein constructed above, we might design a family of such synthetic proteins with the same glue, but random side chains of the frame/shell type above. Then a protein which will have frames whose strands have length 1 corresponding to edges of Γ where $J = -1$ or $J = 0$, length 2 for edges where $J = 1$, and matching lengths for the shells, will fold such that its minimum energy conformation enables to read off the spin values for the given instance of GSG, as specified in the last part of the proof of Theorem 1.

One approach for making the protein and ensuing experiments more realistic, would be to initiate protein-cooperation between physical or structural chemists, biologists and computer scientists or mathematicians, similarly to the DNA-cooperation that already exists in the area of the human genome and DNA-computing.

Conclusion

We have encoded the NP-complete spin glass problem in terms of a minimum energy model of protein folding, such that an instance of the former is a YES-instance if and only if its encoding is a YES-instance of the latter. We have also seen that even the problem of binary side chain foldings of proteins is NP-complete. The protein constructed in the proof contains only three amino acids and approximates true folding.

It would be of interest to encode other NP-complete problems in terms of protein folding. It would also be interesting to do some experimental work with real proteins, aimed at using them for speeding up computations. Whereas each element in DNA computations is well-understood and involves a simple slow computation, but the high parallelism of their great quantity is exploited for simulating the operation of a nondeterministic Turing machine, each protein is a "black box" computer which may have higher computational power than a DNA strand.

Acknowledgment. I thank Ron Unger, whose criticism on an early draft led me to an improved protein construction.

References

1. [Anf1973] C.B. Anfinsen [1973], Principles that govern the folding of protein chains, *Science* **181**, 223–230.

2. [ArL1997] S. Arora and C. Lund [1997], Hardness of Approximations, Ch. 10 in: *Approximation Algorithms for NP-Hard Problems* (D.S. Hochbaum, ed.), PWS Publishing Company, Boston, pp. 399–446.

3. [Bar1982] F. Barahona [1982], On the computational complexity of Ising spin glass models, *J. Phys. A: Math. Gen.* **15**, 3241–3253.

4. [BMRU1982] F. Barahona, R. Maynard, R. Rammal and J.P. Uhry [1982], Morphology of ground states of two-dimensional frustration model, *J. Phys. A: Math. Gen.* **15**, 673–699.

5. [Bax1982] R.J. Baxter [1982], *Exactly Solved Models in Statistical Mechanics*, Academic Press, London.

6. [BBBSS1992] C.H. Bennett, F. Bessette, G. Brassard, L. Salvail and J. Smolin [1992], Experimental quantum cryptography, *J. Crypto.* **5**, 3–28.

7. [BB1989] C.H. Bennett and G. Brassard [1989], The dawn of a new era for quantum cryptography: the experimental prototype is working!, *Assoc. Comput. Mach. SIGACT News* **20**(4), 78–82.

8. [BBCS1991] C.H. Bennett, G. Brassard, C. Crépeau and M.-H. Skubiszewska [1991], Practical quantum oblivious transfer, *Proc. Crypto.* **91**, 351–366.

9. [BBM1992] C.H. Bennett, G. Brassard and N.D. Mermin [1992], Quantum cryptography without Bell's theorem, *Phys. Rev. Lett.* **68**, 557–559.

10. [BBD1995] S.F. Betz, J.W. Bryson and W.F. DeGrado [1995], Native-like and structurally characterized designed α-helical bundles, *Current Opinion in Structural Biology* **5**, 457–463.

11. [BMRU1980] I. Bieche, R. Maynard, R. Rammal and J.P. Uhry [1980], On the ground states of the frustration model of a spin glass by a matching method of graph theory, *J. Phys. A: Math. Gen.* **13**, 2553–2567.

12. [Bra1988] G. Brassard [1988], *Modern Cryptology*, Lecture Notes in Computer Science, Springer-Verlag, Vol. 325, New York.

13. [BC1990] G. Brassard and C. Crépeau [1990], Quantum bit commitment and coin tossing protocols, *Proc. Crypto.* **90**, 49–61.

14. [CDS1996] M.H.J. Cordes, A.R. Davidson and R.T. Sauer [1996], Sequence space, folding and protein design, *Current Opinion in Structural Biology* **6**, 3–10.

15. [DDL1994] J. Desmet, M. De Maeyer and I. Lasters [1994], The "dead-end elimination" theorem: a new approach to the side-chain packing problem, in: *The Protein Folding Problem and Tertiary Structure Prediction* (K. Merz, Jr. and S. Le Grand, eds.), Birkhäuser, Boston, 1994, Ch. 10, pp. 308–337.

16. [Deu1985] D. Deutsch [1985], Quantum theory, the Church-Turing principle and the universal quantum computer, *Proc. Roy. Soc. London A* **400**, 97–117.

17. [Deu1989] D. Deutsch [1989], Quantum computational networks, *Proc. Roy. Soc. London A* **425**, 73–90.

18. [Fra1990] A.S. Fraenkel [1990], Deexponentializing complex computational mathematical problems using physical or biological systems, Tech. Report CS90–30, Dept. of Appl. Math. and Computer Science, Weizmann Inst. of Science.

19. [Fra1993] A.S. Fraenkel [1993], Complexity of protein folding, *Bull. Math. Biol.* **55**, 1199–1210.

20. [GaJ1979] M.R. Garey and D.S. Johnson [1979], *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA.

21. [GiK1990] L.M. Gierasch and J. King (eds.) [1990], *Protein Folding: Deciphering the Second Half of the Genetic Code*, Amer. Assoc. Adv. Sci., Washington, D.C.

22. [HaI1996a] W.E. Hart and S. Istrail [1996], Fast protein folding in the hydrophobic-hydrophilic model within three-eights of optimal, *J. Computational Biology* **3**, 53–96.

23. [HaI1996b] W.E. Hart and S. Istrail [1996], Invariant patterns in crystal lattices: implications for protein folding algorithms (Extended Abstract), Proc. 1996 Combinatorial Pattern Matching, Lecture Notes in Computer Science No. 1075, pp. 288-303, Springer Verlag 1996.

24. [HaI1997] W.E. Hart and S. Istrail [1997], Robust proofs of NP-hardness for protein folding: general lattices and energy potentials, *J. Computational Biology* **4**, 1–22.

25. [HaI≥1997] W.E. Hart and S. Istrail [≥1997], Lattice and off-lattice side chain models of protein folding: linear time structure prediction better than 86% of optimal, *J. Computational Biology*, to appear.

26. [HBDPC1997] P.M. Harrison, P. Bamborough, V. Daggett, S.P. Prusiner and F.E. Cohen [1997], The prion folding problem, *Current Opinion in Structural Biology* **7**, 53–59.

27. [JaW1983] J. Janin and S.J. Wodak [1983], Structural domains in proteins and their role in the dynamics of protein function, *Prog. Biophys. Mol. Biol.* **42**, 21–78.

28. [Joh1983] D.S. Johnson [1983], The NP-completeness column: an ongoing guide, *J. Algorithms* **4**, 87–100.

29. [Joh1997] G. Johnson [1997], Designing life: proteins 1, computers 0, The New York Times (Science Times), March 25, 1997.

30. [KSXBH1993] S. Kamtekar, J.M. Schiffer, H. Xiong, J.M. Babik and M.H. Hecht [1993], Protein design by binary patterning of polar and nonpolar amino acids, *Science* **262**, 1680–1685.

31. [LeL1969] M. Levitt and S. Lifson [1969], Refinement of protein conformations using a macromolecular energy minimization procedure, *J. Mol. Biol.* **46**, 269–279.

32. [LeS1988] M. Levitt and R. Sharon [1988], Accurate simulation of protein dynamics in solution, *Proc. Natl. Acad. Sci. USA* **85**, 7557–7561.

33. [NgM1992] J.T. Ngo and J. Marks [1992], Computational complexity of a problem in molecular-structure prediction, *Protein Engineering* **5**, 313–321.

34. [NMK1994] J.T. Ngo, J. Marks and M. Karplus [1994], Computational complexity, protein structure prediction, and the Levinthal paradox, in: *The Protein Folding Problem and Tertiary Structure Prediction* (K. Merz, Jr. and S. Le Grand, eds.), Birkhäuser, Boston, 1994, Ch. 14, pp. 433–506.

35. [PaP1996] M. Paterson and T. Przytycka [1996], On the complexity of string folding, *Discr. Appl. Math.* **71**, 217–230.

36. [Pri1979] P.L. Privalov [1979], Stability of proteins, *Adv. Protein Chem.* **33**, 167–241.

37. [Pri1982] P.O. Privalov [1982], Stability of proteins: proteins which do not present a single cooperative system, in *Advances in Protein Chemistry* **35**, 1–104.

38. [Ree1988] G.N. Reeke, Jr. [1988], Protein folding: computational approaches to an exponential time problem, in: *Annual Reviews of Computer Science* **3**, 59–84, Annual Reviews, Inc.

39. [RoS1996] N. Robertson and P.D. Seymour [1996], Graph minors. XV. Giant steps, *J. Combin. Theory* Ser.B **68**,112–148, 1996.

40. [Tra1996] J. Traub [1996], On reality and models, in: *Boundaries and Barriers: On the Limits to Scientific Knowledge* (J.l. Casti and A. Karlqvist, eds.), Addison-Wesley, 1996, pp. 238-254.

41. [UnM1993] R. Unger and J. Moult [1993], Finding the lowest free energy conformation of a protein is a NP-hard problem: proof and implications, *Bull. Math. Biol.* **55**, 1183–1198.

42. [WaC1986] S.A. Wasserman and N.R. Cozzarelli [1986], Biochemical topology: applications to DNA recombination and replication, *Science* **232**, 951–960.

43. [Wel1993] D.J.A. Welsh [1993], The complexity of knots, *Ann. Disc. Math.* **55**, 159–171.

44. [WeH1995] M.W. West and M.H. Hecht [1995], Binary patterning of polar and nonpolar amino acids in the sequences and structures of native proteins, *Protein Science* **4**, 2032–2039.

45. [Wie1983] S. Wiesner [1983], Conjugate coding, *Assoc. Comput. Mach. SIGACT News* **15**(1), 78–88 (manuscript written about 1970).

Department of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel

fraenkel@wisdom.weizmann.ac.il `http://www.wisdom.weizmann.ac.il/~fraenkel`