Computational Complexity of a Problem in Molecular-Structure Prediction

Computational Complexity of a Problem in Molecular-Structure Prediction

Running title
Complexity of Molecular-Structure Prediction

Key words
intractability
NP-completeness
peptide backbone conformation
potential-energy minimization
protein-structure prediction

J. Thomas Ngo[1]
Committee on Higher Degrees in Biophysics
Harvard University
Cambridge, MA 02138
USA

Joe Marks[2]
Center for Research in Computing Technology
Harvard University
Cambridge, MA 02138
USA

June 25, 1991
Revised February 14, 1992
Appeared in *Protein Engineering* **5**(4):313–321 (1992)

[1] To whom correspondence should be addressed.
[2] Current address: DEC CRL, One Kendall Square, Cambridge, MA 02139, USA.

Abstract

The computational task of protein-structure prediction is believed to require exponential time, but previous arguments as to its intractability have taken into account only the size of a protein's conformational space. Such arguments do not rule out the possible existence of an algorithm, more selective than exhaustive search, that is efficient and exact. (An *efficient* algorithm is one that is guaranteed, for all possible inputs, to run in time bounded by a function polynomial in the problem size. An *intractable* problem is one for which no efficient algorithm exists.) Questions regarding the possible intractability of problems are often best answered using the theory of NP-completeness. In this treatment we show the NP-hardness of two typical mathematical statements of empirical potential-energy-function minimization for macromolecules. Unless all NP-complete problems can be solved efficiently, these results imply that a function-minimization algorithm can be efficient for protein-structure prediction only if it exploits protein-specific properties that prohibit the simple geometric constructions that we use in our proofs. Analysis of further mathematical statements of molecular-structure prediction could constitute a systematic methodology for identifying sources of complexity in protein folding, and for guiding development of predictive algorithms.

Introduction

A major goal in structural molecular biology is to find a computational procedure for determining the minimum-energy conformation of a given polypeptide chain. This problem has evaded exact solution by current methods. Clearly, exhaustive search of a protein's conformational space is out of the question: the size of the space is exponential in the size of the molecule, and contemporary hardware falls many orders of magnitude short for even the smallest proteins (Reeke, 1988). From these facts and the failure of years of research to produce an efficient algorithm (one that is guaranteed to terminate correctly in polynomial time for all possible inputs), it has been widely inferred that the problem is intractable (*i.e.*, that no efficient algorithm for solving the problem exists). However, the validity of this inference has never been proven (see Appendix A). Without suitable analysis, we cannot rule out the existence of a technique, necessarily more selective than exhaustive search, for efficient global minimization of the empirical potential-energy function.

Rigorous tools for exploring the possible intractability of a problem are provided by the results and techniques of computational complexity theory—in particular, the theory of NP-completeness (Garey and Johnson, 1979; Lewis and Papadimitriou, 1981). (Some rudimentary definitions may be found in Appendix B.) Physical models have seldom been studied in terms of their computational complexity. One notable exception is the finding that several Ising spin glass models are NP-hard (Barahona, 1982). In this paper we present initial results in such a treatment of protein-structure prediction. We have two motivations for undertaking this analysis. Like widely held beliefs in many other fields, the intractability of the protein-folding problem is, strictly speaking, merely an opinion; a rigorous inquiry into its validity is of intrinsic intellectual interest. On a more utilitarian level, continued formal analysis may permit us to understand better the sources of the problem's complexity, and thus influence the development of predictive algorithms.

This type of analysis is of indirect relevance to an argument attributed to Levinthal (Levinthal, 1968), which suggests that *in nature* a protein molecule cannot possibly sample all conformational states as it folds. By contrast, our approach is intended to apply solely to *algorithms*—we treat a mathematical model of protein structure as given, and determine the computational complexity of using it for structure prediction. The exact relationship between the computational complexity of predictive algorithms and the behavior of molecules is an interesting and important issue, but we do not address it here.

Our strategy is typical of NP-completeness proofs. The non-trivial part of the proof is to demonstrate the existence of an efficient transformation from some known NP-complete problem to the one in question, *i.e.*, to show that a known NP-complete problem is *reducible* to it. The principle of reducibility provides a formal measure of the problems' relative difficulty: were an efficient algorithm for the problem in question to exist, it could be combined with the demonstrated transformation to form an efficient solution to the known NP-complete problem. Informally, the reduction shows that the problem in question is at least as difficult as the known NP-complete problem. The existence of such a hybrid algorithm would imply the heretofore unproved (and very unlikely) equivalence of the computational classes P and NP—the existence of an efficient solution to every NP-complete problem. (See Appendix B for more details.)

For clarity we have chosen to demonstrate the reduction in separate stages. We first define a discrete computational problem that we call DIAMOND LATTICE PATH (DLP), and prove it to be NP-complete by reduction from a known NP-complete problem called PARTITION (Garey and Johnson, 1979). Corollary to this intermediate result is the NP-hardness of various forms of global potential-energy minimization. We demonstrate this corollary reduction explicitly for two commonly encountered energy-minimization tasks.

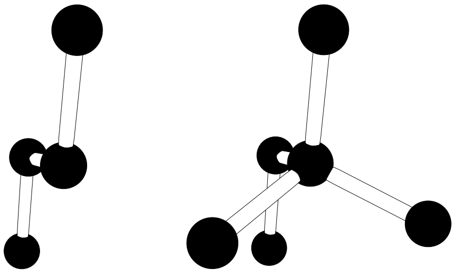

Figure 1: Preferred conformations of butane (left) and neohexane when in isolation. Hydrogens are omitted for clarity. By symmetry, the potential energy of neohexane is invariant with respect to deformation by torsion of 120° about the central bond.

Results

DIAMOND LATTICE PATH is NP-complete DIAMOND LATTICE PATH (DLP) is an abstract problem inspired by the geometry of aliphatic chains. A discussion of the relevance of DLP to protein folding is reserved for a later section; here we concentrate on the relationship between DLP and alkanes. An alkane is an n-carbon chain of the form $CH_3(CH_2)_{n-2}CH_3$. Each carbon atom is sp^3 hybridized, *i.e.*, has four tetrahedrally positioned neighbors. Torsion about each bond is relatively unhindered at normal temperatures, but each dihedral angle has three preferred values, $\{180^\circ, \pm 60^\circ\}$. The 180° conformation is favored in an isolated molecule of butane ($CH_3CH_2CH_2CH_3$); the terminal carbons are maximally separated due to 1–4 Van der Waals repulsion (Streitwieser and Heathcock, 1976). When the two hydrogens on the third carbon are replaced by methyl groups, giving neohexane ($CH_3CH_2C(CH_3)_3$), by symmetry all three values are equally favored (see Figure 1).

DLP corresponds to structure prediction for idealized n-carbon chains in which the tetrahedral bond geometry is taken to be exact, and one can choose arbitrarily which dihedrals along the chain will favor the 180° configuration, and which will have equal optimal energy for the values $\{180^\circ, \pm 60^\circ\}$. We shall refer to the latter as "threefold" dihedrals. Given these idealized properties, the permissible bond-length, angle, and dihedral values are exactly those found in a diamond crystal (Kittel, 1976), which is composed entirely of sp^3 carbons. Thus, the carbon backbone follows a path that can be embedded in a regular diamond lattice (see Figure 2), with certain restrictions on the form of the embedding. We therefore state our discrete problem in terms of paths in a diamond lattice, rather than as function minimization.

Define the diamond lattice D to be the infinite set of points in three-dimensional space that are occupied by the carbon atoms in a diamond crystal. Define $\mathcal{D}$ to be the set of paths that can be traversed along bonds in a diamond lattice. That is, a sequence of points $(m_0, m_1, \ldots, m_N)$, each in D, is said to be a path in $\mathcal{D}$ if and only if every pair of consecutive points in the sequence is a pair of nearest neighbors. For a given path $(m_0, m_1, \ldots, m_N)$ we shall refer to the vector $m_i - m_{i-1}$ as the ith *bond,* or simply δ_i. The task in DLP is to determine the existence of a path in $\mathcal{D}$ that satisfies given turn restrictions and endpoint constraints.

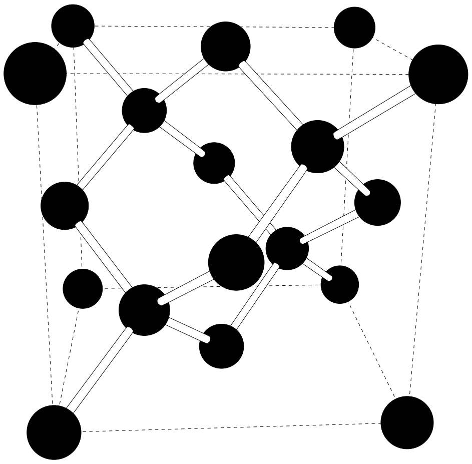

Figure 2: Unit cell of the diamond lattice D. Each point in D has four nearest neighbors. With a convenient choice of distance units and orientation, D is the set of points m such that $m = m' + 4(i, j, k) + (1, 1, 1)l$, where $m' \in \{(0,0,0), (0,2,2), (2,0,2), (2,2,0)\}$, $l \in \{0, 1\}$, and $i, j, k \in Z$. (Knowledge of these coordinates is not essential to an understanding of the proof.)

DIAMOND LATTICE PATH (DLP)
INSTANCE: A positive integer N; paths (P_0, P_1, P_2) and (Q_0, Q_1, Q_2), both in $\mathcal{D}$; and a set $I \subseteq \{2, 3, \ldots, N-1\}$.
QUESTION: Is there a path $(m_0, m_1, \ldots, m_N)$ in $\mathcal{D}$, such that $(m_0, m_1, m_2) = (P_0, P_1, P_2)$, $(m_{N-2}, m_{N-1}, m_N) = (Q_2, Q_1, Q_0)$, and $\delta_{i-1} = \delta_{i+1}$ for every $i \in (\{2, 3, \ldots, N-1\} - I)$?

Dihedrals about bonds whose indices appear in I are threefold. The size of a DLP instance description is linear in $|I|$, the number of threefold dihedrals along the chain, not in its length N.

The proof that DLP is NP-complete proceeds in two steps. The first step is to show that DLP is in the class NP. DLP $\in$ NP because any affirmative solution instance, expressed as m_0, m_1, and m_2, followed by a list of the values chosen for each threefold symmetric dihedral, can be verified in time polynomial in $|I|$ by computing the positions of m_{N-2}, m_{N-1}, and m_N, and checking the endpoint conditions. The second step is to show that a known NP-complete problem, PARTITION, is polynomial-time reducible to DLP:

PARTITION
INSTANCE: A finite set $A = \{a_1, a_2, \ldots, a_{|A|}\}$, and a size $s(a) \in Z^+$ for each $a \in A$. (Z^+ is the set of positive integers.)
QUESTION: Is there a subset $A' \subseteq A$ such that $\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a)$?

We demonstrate that PARTITION is efficiently reducible to DLP by describing a polynomial-time procedure for converting any instance of PARTITION into some instance of DLP. The essence of this procedure is to divide the path $(m_0, m_1, \ldots, m_N)$ into subsequences whose

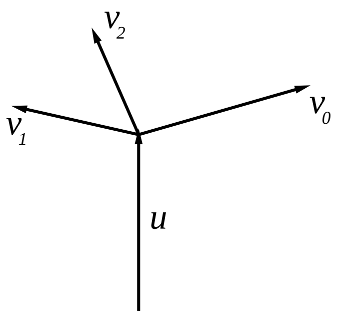

Figure 3: Redundant basis vectors u, v_0, v_1 and v_2 used to specify translations in the diamond lattice.

lengths correspond to the integers in the PARTITION instance, and to choose the endpoint conditions so that they are satisfied if and only if the PARTITION instance can be answered in the affirmative. The endpoints are chosen so that any chain connecting them must be fully extended. This yields instances of DLP for which simple properties may be deduced and exploited.

The procedure is to set the parameters in the instance description as follows. As in Figure 3, define vectors $\{-u, v_0, v_1, v_2\}$ to specify, in any order, the translations from the origin to its four nearest neighbors in D. These will serve as a redundant basis set; their precise coordinates will not be required since only their symmetry properties are used in the proof. Set the endpoint conditions:

$$\begin{array}{rclrcl} P_0 & = & \mathbf{0} & Q_2 & = & P_1 + \Delta^0 \\ P_1 & = & v_0 & Q_1 & = & P_1 + \Delta^0 + u \\ P_2 & = & v_0 + u & Q_0 & = & P_1 + \Delta^0 + u + v_0 \\ & & & \Delta^0 & = & (2u + v_0 + v_1)B \end{array}$$

and the path length and list of threefold dihedrals:

$$\begin{array}{rcl} B & = & \frac{1}{2}\sum_{a \in A} s(a) \\ N & = & 4B + 3 \\ I & = & \{2, 2s(a_1) + 2, 2s(a_1) + 2s(a_2) + 2, \ldots, 4B + 2\}. \end{array}$$

Note that B is a whole number; if $\sum_{a \in A} s(a)$ is not even, then trivially no partition of A can meet the requirements.

This construction takes only time polynomial in the size of the PARTITION instance (an appropriate measure for the size of an instance of PARTITION is $|A| \log B$, since $|A|$ integers must be stored, and the number of bits required to store each integer is bounded above by $\log B$). We claim that an instance of DLP so constructed will have an affirmative answer if and only if the original PARTITION problem has an affirmative answer.

The paths $\mathcal{D}$ are such that each bond must lie in one of eight directions; namely, $\delta_i \in \{\pm u, \pm v_0, \pm v_1, \pm v_2\}$ for every $1 \leq i \leq N$. We fix m_0, m_1, and m_2 at P_0, P_1, and P_2, respectively, and take Δ to be the displacement from m_1 to m_{N-2}. The proof of our claim consists of

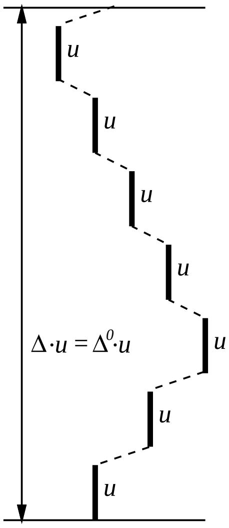

Figure 4: Schematic representation of a path projected onto some plane containing u. The endpoint conditions constructed for a given instance of PARTITION are such that the path cannot span the required distance in u unless alternating bonds point along u.

determining the conditions under which there exists some sequence of bond directions that satisfies the requirement $\Delta = \Delta^0$. We show first that alternating bonds in the sequence must lie in the u direction, and that none of the remaining bonds can lie in the v_2 direction.

Consider $\Delta \cdot u$. By symmetry, the projections of the directions $\{\pm u, \pm v_0, \pm v_1, \pm v_2\}$ onto u have only four possible values. In decreasing order, they are:

$$\begin{array}{cr} u \cdot u & 1 \\ v_0 \cdot u = v_1 \cdot u = v_2 \cdot u & \frac{1}{3} \\ -v_0 \cdot u = -v_1 \cdot u = -v_2 \cdot u & -\frac{1}{3} \\ -u \cdot u & -1 \end{array}$$

(Only the ordering is important; the values themselves are not.) The required displacement in the positive u direction is $\Delta^0 \cdot u = (2u \cdot u + v_0 \cdot u + v_1 \cdot u)B$. Since no two contiguous bonds can both equal u, this is the maximum attainable, and is realized only if $\delta_i = u$ for $i \in \{2, 4, 6, \ldots, 4B\}$, while $\delta_i \in \{v_0, v_1, v_2\}$ for $i \in \{3, 5, 7, \ldots, 4B+1\}$. (See Figure 4.) For brevity we shall refer to the former bonds as the u bonds, and to the latter, as the v bonds.

Assign to the sets V_0, V_1, and V_2 all v bonds between m_1 and m_{N-2} that point in the directions v_0, v_1, and v_2, respectively. Because the number of v bonds between those two points is $2B$, it follows that $|V_0| + |V_1| + |V_2| = 2B$. Consider $\Delta \cdot w$, where w is the projection of $v_0 + v_1$ onto the plane orthogonal to u. (That is, $w = (v_0 + v_1) - [(v_0 + v_1) \cdot u]u$.) By the definition of w, the u bonds do not contribute, so that $\Delta \cdot w = |V_0| v_0 \cdot w + |V_1| v_1 \cdot w + |V_2| v_2 \cdot w$. But by symmetry, $v_0 \cdot w = v_1 \cdot w > v_2 \cdot w$. Since the required displacement is $\Delta^0 \cdot w = (v_0 \cdot w + v_1 \cdot w)B$, it must be that $|V_2| = 0$. (See Figure 5.)

Thus, $\Delta = (2B)u + |V_0|v_0 + |V_1|v_1$. To have $\Delta = \Delta^0$ it is required that $|V_0| = B$ and $|V_1| = B$. However, an arbitrary assignment of v bonds to V_0 and V_1 is not permitted since

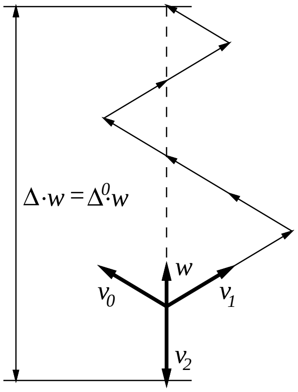

Figure 5: Schematic representation of a path projected onto some plane orthogonal to u. The endpoint conditions constructed for a given instance of PARTITION are such that the path cannot span the required distance along w unless no bond points along v_2.

changes in direction between adjacent v bonds δ_{i-1} and δ_{i+1} are possible only for $i \in I$. The values assigned to I in this instance of DLP divide the sequence of v bonds into subsequences of lengths $s(a_1), s(a_2), \ldots, s(a_{|A|})$; and within each subsequence all v bonds are equal. By inspection of Figure 6, it can be seen that there is a path in $\mathcal{D}$ that satisfies the endpoint and directional constraints if and only if the original PARTITION problem can be answered in the affirmative. This completes the proof that DLP is NP-complete.

ECPSP energy minimization is NP-hard We now demonstrate reductions from DLP to two more general problems, in which attention is focused upon global minimization of an empirical potential-energy function U. We phrase each as a *decision* problem, in which the objective is to find out whether the potential function has some value below a given threshold. A decision problem that is derived from an optimization problem is always trivially reducible to it, since the existence of an efficient solution to the optimization problem would immediately imply the existence of an efficient solution to the decision problem. Before we demonstrate the reductions, we address three ancillary technical points, all of which pertain to the possible occurrence of irrational numbers among the input and output values in particular problem instances.

1. The decision problems described below may not be in NP. For an algorithm to verify a candidate solution in polynomial time, it must use arithmetic of limited precision. To discover whether these problems are in NP would necessitate a sophisticated error analysis beyond the scope of this paper. The proofs given here therefore imply NP-hardness, but not NP-completeness (see Appendix B).

2. Even if the decision problems are in NP, the corresponding optimization problems might not be.

3. For the transformations from DLP to be efficient, the sizes of the problem instances that they produce must satisfy certain bounds. This point is addressed below.

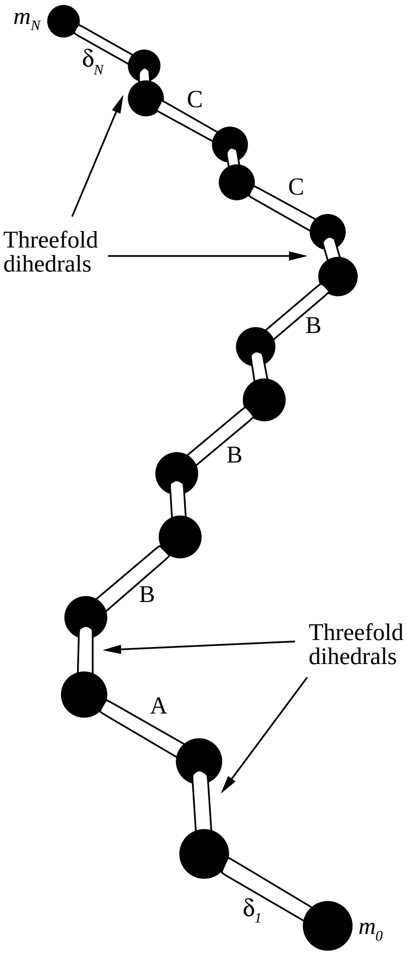

Figure 6: Sample proof construction, for a case in which the corresponding PARTITION problem has been answered in the affirmative. The values of $s(a)$ are 1, 3 and 2, and they map onto the subsequences of v bonds labeled A, BBB and CC.

The first problem, ENDPOINT CONSTRAINED POLYMER STRUCTURE PREDICTION (ECPSP), is encountered when predicting the structure of an internal backbone segment given the remainder of the protein's structure. This type of situation is common in homology modeling (Bruccoleri et al., 1988, for example), in which most of a protein's structure is initially assumed to be identical to that of another, homologous, protein whose structure has already been determined experimentally. ECPSP is the task of predicting the structure of a backbone subsegment whose conformation could not be taken from the known structure. Obviously, the primary benefit of homology modeling is the drastic reduction in the number of degrees of freedom; these segments are much shorter than the protein as a whole. Further benefit accrues from the requirement that the endpoints of the backbone subsegment meet the remainder of the protein at known positions in space and with restricted orientations. These extra conditions, sometimes known as "loop-closure" constraints, remove three translational and three rotational degrees of freedom from the space of possible backbone conformations (Gō and Scheraga, 1970).

ENDPOINT CONSTRAINED
POLYMER STRUCTURE PREDICTION (ECPSP)
INSTANCE: An $(N+1)$-tuple of atoms, $(m_0, m_1, m_2, \ldots, m_N)$, each to be situated in three-dimensional space; six fixed points, P_0, P_1, P_2, Q_0, Q_1, and Q_2; an equilibrium bond length l_b^0 and positive coefficient K_b^{bond} for each adjacent pair of atoms, $b = (m_i, m_{i+1})$, in the tuple; an equilibrium angle θ_a^0 and positive coefficient K_a^{angle} for each adjacent triplet of atoms, $a = (m_i, m_{i+1}, m_{i+2})$; an equilibrium dihedral angle ϕ_d^0, positive integer n_d, and positive coefficient K_d^{dihedral} for each adjacent quadruplet of atoms, $d = (m_i, m_{i+1}, m_{i+2}, m_{i+3})$; and an energy bound, U^0.

QUESTION: Given that the atoms m_0, m_1, m_2, m_{N-2}, m_{N-1}, and m_N must be positioned at points P_0, P_1, P_2, Q_2, Q_1, and Q_0, respectively, are there values for the geometric parameters l_b, θ_a and ϕ_d, that cause the potential function

$$\begin{aligned} U \;&=\; \sum_b K_b^{\text{bond}}(l_b - l_b^0)^2 + \sum_a K_a^{\text{angle}}(\theta_a - \theta_a^0)^2 + \\ &\quad\; \sum_d K_d^{\text{dihedral}}(1 - \cos\left[n_d(\phi_d - \phi_d^0)\right]) + \text{non-local terms} \end{aligned}$$

to have a value not exceeding U^0?

Some aspects of the problem statement require explanation:

- The expression for U is based upon a typical form of the empirical potential-energy function (Brooks et al., 1988). For clarity we have not written out non-local terms, *e.g.*, terms arising from Van der Waals forces, electrostatic interactions, and hydrogen bonds. When using a hypothetical ECPSP algorithm to solve DLP problems, the coefficients of these non-local terms would be set to zero.
- To simplify the presentation we have given a straightforward instance description for ECPSP. However, for the reduction to be efficient, the size of the ECPSP instance

description must be provably polynomial in the size of the corresponding DLP instance description, and this is not the case because they are $O(N)$ and $O(|I|)$, respectively. This is rectified by an $O(|I|)$ encoding scheme for the former, in which default values K^{bond}, K^{angle}, K^{dihedral}, l^0, θ^0, ϕ^0, and n apply uniformly across the chain, and non-default parameter values are stored as ordered pairs in which the first element identifies a particular constant and the second element specifies its value. In the reduction from DLP to ECPSP, the default value n of the dihedral periodicity is set to 1, and all of the ordered pairs are used to override values of n_d along the chain. The size of the instance description thus encoded is, as expected, proportional to the number of dihedrals with multiple optima, and not to the length of the chain. Similar compact encoding schemes are easily constructed for molecules in which each repeated monomeric unit contributes more than one atom to the backbone.

- The instance descriptions produced by the transformation from DLP to ECPSP must contain only parameter values that can be represented using finite storage. This condition is satisfied if bond lengths are represented in units of $\sqrt{3}$, angles are represented in units of arccos $(-\frac{1}{3})$, and endpoints are stored using the basis vectors $\{-u, v_0, v_1, v_2\}$.

We prove that ECPSP is NP-hard by showing a polynomial-time reduction from DLP. The transformation from an arbitrary instance of DLP to an instance of ECPSP is straightforward: the hypothetical algorithm for ECPSP is simply configured, through appropriate choice of parameters, to model the idealized alkane in the DLP instance. Namely, l_b^0 is set to 1 ($\times\sqrt{3}$) for all b; θ_a^0 is set to 1 (× arccos $(-\frac{1}{3})$) for all a; ϕ_d^0 is set to 180° for all d; n_d is set to 3 for threefold dihedrals, and to 1 for all others. The endpoints are merely copied; the coefficients K_b^{bond}, K_a^{angle}, and K_d^{dihedral} can take on any positive, non-zero values; and the energy bound U^0 is set to zero. Since all of the terms are non-negative, U is zero if and only if all of its terms are zero. It follows that an instance of ECPSP so constructed will be answered in the affirmative if and only if the corresponding DLP problem has an affirmative answer. Informally, the existence of this reduction shows that ECPSP is a generalization of DLP and therefore cannot be easier to solve.

PSP energy minimization is NP-hard Finally, we introduce the second minimization problem, POLYMER STRUCTURE PREDICTION (PSP). It differs from ECPSP in the lack of endpoint constraints and in the obligatory presence in U of at least one non-local term that is a function of interatomic distance with a minimum at some adjustable radius:

POLYMER STRUCTURE PREDICTION (PSP)
INSTANCE: An $(N+1)$-tuple of atoms, $(m_0, m_1, m_2, \ldots, m_N)$, each to be situated in three-dimensional space; an equilibrium bond length l_b^0 and positive coefficient K_b^{bond} for each adjacent pair of atoms, $b = (m_i, m_{i+1})$, in the tuple; an equilibrium angle θ_a^0 and positive coefficient K_a^{angle} for each adjacent triplet of atoms, $a = (m_i, m_{i+1}, m_{i+2})$; an equilibrium dihedral angle ϕ_d^0, positive integer n_d, and positive coefficient K_d^{dihedral} for each adjacent quadruplet of atoms, $d = (m_i, m_{i+1}, m_{i+2}, m_{i+3})$; a positive coefficient $K_{ij}^{\text{non-local}}$ and equilibrium radius r_{ij}^0 for each pair of atoms $i > j$ in the tuple; and an energy bound, U^0.

QUESTION: Are there values for the internal coordinates l_b, θ_a and ϕ_d, that cause the potential function

$$\begin{aligned} U &= \sum_b K_b^{\text{bond}}(l_b - l_b^0)^2 + \sum_a K_a^{\text{angle}}(\theta_a - \theta_a^0)^2 + \\ &\quad \sum_d K_d^{\text{dihedral}}(1 - \cos\left[n_d(\phi_d - \phi_d^0)\right]) + \\ &\quad \sum_{i>j} K_{ij}^{\text{non-local}} f(r_{ij}/r_{ij}^0) + \text{other non-local terms} \end{aligned}$$

to have a value not exceeding U^0? (The variable r_{ij} is the distance between atoms i and j. For a problem to qualify as an instance of PSP, the dimensionless function $f(x)$ must have a unique global minimum at $x = 1$.)

Since PSP is neither a generalization nor a specialization of ECPSP, its status with respect to NP-hardness must be established independently. Fortunately, its reduction from PARTITION is very similar in principle to that of ECPSP. This reduction is most easily understood in terms of DLP′, which is the subset of DLP instances that can arise via reduction from PARTITION instances. Trivially, DLP′ is NP-complete.

We again set U^0 equal to the sum of the individual terms' global minima, so that the only admissible conformations are those for which all terms in U achieve their global minima simultaneously. The constructed backbone consists of two regions. The "variable" region consists of atoms $(m_0, m_1, m_2, \ldots, m_\nu)$, where ν is equal to N from the DLP instance, and it has parameters identical to those of the chain constructed in the reduction of DLP to ECPSP. From that reduction, we know that in any admissible conformation, the variable region can be oriented to follow a path in $\mathcal{D}$, and the non-threefold dihedrals are in the 180° configuration. Without affecting the outcome of the PSP question we may restrict our attention to configurations that are oriented so that atoms $m_{\nu-2}$, $m_{\nu-1}$, and m_ν are positioned at points Q_2, Q_1, and Q_0, respectively.

It remains to require that in any admissible configuration, the atoms m_0, m_1, and m_2 lie at points P_0, P_1, and P_2. These conditions are realized by non-local energy terms in conjunction with a second region, the "scaffolding." The default value of the periodicity, $n = 1$, is not overridden anywhere in this region; thus, it has a unique zero-energy configuration which we are free to determine in the obvious manner, by setting the remaining geometric parameters. The requirements of a valid reduction leave a considerable amount of latitude in choosing this configuration. Here we state only the essential features of the reduction:

- All but a selected few of the coefficients $K_{ij}^{\text{non-local}}$ are set to zero. Each remaining $K_{ij}^{\text{non-local}}$ is set to some positive number, so that the corresponding term achieves its global minimum if and only if $r_{ij} = r_{ij}^0$.
- The distance constraints must be sufficient to rule out all configurations except those for which the endpoint constraints in question are satisfied. In one possible scheme, four distance constraints are used to determine the position of each atom in $\{m_0, m_1, m_2\}$: one to remove each spatial degree of freedom and one more to break symmetry. Specifically,

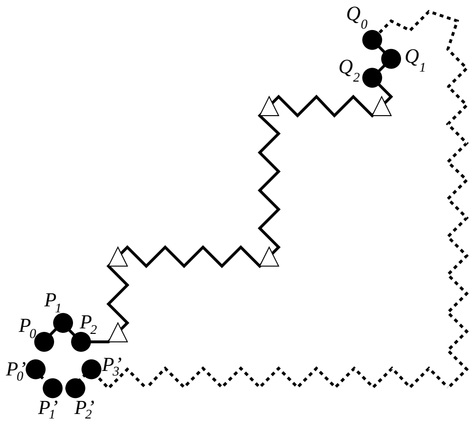

Figure 7: Schematic representation of one possible transformation from DLP′ to PSP. As in the transformation from DLP to ECPSP, the task is to find a set of values for the threefold dihedrals (marked by triangles) that causes the original chain (solid lines) to span from (Q_0, Q_1, Q_2) to (P_2, P_1, P_0). The relative positioning of those six endpoint atoms is determined by the "scaffolding" (broken lines), which is flexible but has a unique zero-energy conformation.

we choose geometric parameters such that in a zero-energy conformation the atoms m_N, m_{N-1}, m_{N-2}, and m_{N-3} must lie at four points P_0', P_1', P_2', and P_3' which are not coplanar, and among which no three are collinear. We then set the equilibrium distances r_{ij}^0 for the twelve atom pairs in $\{m_0, m_1, m_2\} \times \{m_N, m_{N-1}, m_{N-2}, m_{N-3}\}$ equal to the distances in $\{P_0, P_1, P_2\} \times \{P_0', P_1', P_2', P_3'\}$ (see Figure 7.) Alternatively, it may be shown that just three distance constraints and one reference atom in the scaffolding are sufficient: the atoms m_0, m_1, and m_2 are already known to lie at lattice points in any admissible conformation, and the role of each distance constraint is to break a symmetry.

- The scaffolding need not follow a path in $\mathcal{D}$. However, in keeping with an earlier technical point, it must be possible to specify the scaffolding in polynomial space. In fact, the scaffolding can be specified in constant space by default encoding.

Thus, we have shown that were an efficient, general algorithm for PSP to exist, it could be used to solve arbitrary instances of DLP′ in polynomial time. Since DLP′ is known to be NP-complete, it follows that PSP is NP-hard.

Discussion

Scope of results In the discipline of computer science, establishing the intrinsic difficulty of a problem—typically, proving a problem's NP-hardness, or developing a polynomial-time algorithm for it—is considered an essential first step in its characterization, because the likely utility of any algorithmic technique will depend on the problem's tractability. If the problem might be solved in polynomial time, then it is reasonable to seek exact, efficient algorithms. If the problem is NP-hard, then exact algorithms are likely to be impractical except in the rare cases in which a worst-case exponential-time algorithm has polynomial-time average-case performance over some distribution of expected inputs. (An example is Dantzig's simplex algorithm for linear programming; see Appendix A.) More often, one is forced to consider compromises (*e.g.*, approximation algorithms, probabilistic algorithms, special-case algorithms, and heuristics) that entail well-understood tradeoffs between guarantees of time, accuracy, and certainty (Papadimitriou and Steiglitz, 1982, chapters 16–19).

This standard approach is somewhat complicated in the case of protein-structure prediction; because the problem is of natural origin, it is difficult to write a concise mathematical statement that represents an accurate model, and at the same time is sufficiently specialized, due to restrictions on its parameters, to exclude extraneous problem instances that might affect a worst-case analysis. We have presented an initial analysis of two problems in potential-energy minimization, ECPSP and PSP, which have forms typical of empirical potential-energy models (Brooks et al., 1988) but are more general than necessary. We have shown these to be NP-hard by reduction from PARTITION via DLP and DLP′. We have used the number of multiple-optimum dihedral angles, not chain length, as the measure of problem size. This is appropriate because the sole source of complexity that is identified in our proof construction is the existence of polymodal terms. For a future analysis, especially one in which non-local terms are required to be non-zero, chain length may well be a more appropriate measure.

The methodology pursued here is complementary to one used by Skolnick and co-workers, who carried out a series of Monte Carlo folding and unfolding simulations, also employing a diamond lattice (Sikorski and Skolnick, 1990, and references cited therein). Observations from lattice simulations are measurements of particular systems (albeit artificial ones) and any extrapolation to the behavior of more realistic, continuous models is by inductive reasoning. By contrast, the logic that relates the NP-completeness of DLP and DLP′ to the running time of protein-structure-prediction algorithms is deductive. Our result places severe, unambiguous limitations on the generality of an efficient protein-folding algorithm—even one based on a continuous model: *an algorithm for protein-structure prediction that is based on minimization of a typical empirical potential-energy function cannot be efficient if it is so general that it accommodates DLP′*—barring the unlikely equivalence of the classes P and NP, which would imply the existence of an efficient solution to every NP-complete problem. (To the best of our knowledge, all current energy-minimization algorithms can handle arbitrary instances of PSP, and therefore are general enough to accommodate DLP′.) Furthermore, because all of the geometric parameters used in our reductions from DLP and DLP′ to ECPSP and PSP, respectively, lie well within the ranges permitted for organic molecules, an algorithm that circumvents our result by excluding DLP′ must contain inherent limitations on its generality other than mere restrictions on the range of values that each local geometric parameter is permitted to adopt.

The computational complexity of any problem statement general enough to subsume protein-structure prediction but special enough to exclude DLP′ remains an open question. In practical terms, this means that an energy-minimization algorithm that is efficient for proteins, if one exists, must exploit properties of proteins that are not found in the idealized alkanes used in DLP. In particular, we have not directly addressed the possible effects of compactness requirements. Empirically, proteins are observed to contain only very small cavities; this close packing of the atoms is believed to be accounted for by attractive Van der Waals forces and the effects of hydrophobicity (solvent entropy). The requirement of compactness is known to rule out the vast majority of conformations available to a protein through dihedral-angle variation (Chan and Dill, 1991, review), but its effects on the computational complexity of structure prediction are unknown (see Appendix A). The complexity of an optimization problem containing *obligatory* compactness constraints, and the corresponding possibility of a structure-prediction algorithm that requires compactness criteria to run efficiently, are therefore of interest. The result that we have presented might serve as a baseline for comparative analysis of the effects of this and other protein-specific restrictions.

Epilogue Proofs of NP-hardness are essentially negative results. Nevertheless, complexity results can make a positive contribution by influencing the directions taken by algorithm developers.[1] In the context of backbone-structure prediction, we have suggested how continued systematic analysis of protein-specific restrictions may constitute a mathematically rigorous basis for guiding the otherwise intuitive process of developing useful prediction algorithms. An immediate goal in such an incremental analysis of PSP and ECPSP will be to examine whether their complexities change under the restriction that the coefficients of the non-local potential terms be non-zero.

In addition, such a line of inquiry may be of particular practical value for tasks in protein engineering that appear easier than the general problem, but whose complexities are nonetheless uncertain. For example, recent case studies using simulated annealing (Lee and Subbiah, 1991) have suggested that packing effects may suffice to determine the sidechain conformations in a protein's core. If this is true, then a hard-sphere model containing only short-range effects (*i.e.*, repulsive Van der Waals forces) may suffice for predicting simultaneously the conformations of all of the sidechains in a protein's core. The computational complexity of the packing problem implied by such a model remains to be determined. Because only short-range effects are present, the graph of possible sidechain-sidechain interactions can be known in advance, is sparse, and consists of vertices of low degree. Previous experience—for instance, in graph colorability (Garey and Johnson, 1979, page 191) and cartographic labeling (Marks and Shieber, 1991; Formann and Wagner, 1991)—illustrates that such neighborhood interactions can, on their own, give rise to NP-hardness. On the other hand, many graph problems cease to be NP-hard when restrictions are placed on the nature of the graph, suggesting that this problem of finding a mutually acceptable set of sidechain conformations for a protein may be tractable. Not knowing the computational complexity of sidechain-structure prediction leaves the algorithm developer in the quandary of not knowing whether inexact methods are truly necessary, given the possible existence of a superior exact algorithm.

[1] In independent work, Unger and Moult (Unger and Moult, 1992) have derived a complementary proof that addresses the task of finding the minimum-energy self-avoiding walk on a cubic lattice. The show, by reduction from QUADRATIC ASSIGNMENT (Garey and Johnson, 1979), that the presence of an arbitrary non-local potential energy term is sufficient to render that problem NP-complete.

Acknowledgments

JTN is a student of Martin Karplus, whom we thank for discussions, particularly on the relationship between computational complexity theory and the problem of protein-structure prediction, and for guidance in the preparation of the manuscript. We also thank Leo Caves, Jeff Evanseck, Jeff Evenson, Mark Friedell, Howard Holley, Sandeep Kochhar, Krzysztof Kuczera, Harry Lewis, Harry Mairson, Stuart Shieber, Ron Unger, and Alan Whiting for their comments and discussions. JTN is grateful for a Graduate Fellowship from the Fannie and John Hertz Foundation. This work was supported in part by a National Science Foundation grant to Martin Karplus.

References

A. V. Aho, J. E. Hopcroft, and J. D. Ullman. 1974. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, Massachusetts.

Francisco Barahona. 1982. On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematics and General*, 15:3241–3253.

Charles L. Brooks, III, Martin Karplus, and B. Montgomery Pettitt. 1988. *Proteins*, volume LXXI of *Advances in Chemical Physics*. Wiley, New York.

Robert E. Bruccoleri, Edgar Haber, and Jiri Novotny. 1988. Structure of antibody hypervariable loops reproduced by a conformational search algorithm. *Nature*, 335:564–568, October.

Hue Sun Chan and Ken A. Dill. 1991. Polymer principles in protein structure and stability. *Annual Reviews of Biophysics and Biophysical Chemistry*, 20:447–490.

Stephen A. Cook. 1971. The complexity of theorem-proving procedures. *Proceedings of the Third Annual ACM Symposium on the Theory of Computing*, pages 151–158.

G. B. Dantzig. 1963. *Linear Programming and Extensions*. Princeton University Press, Princeton, New Jersey.

Michael Formann and Frank Wagner. 1991. A packing problem with applications to lettering of maps. In *Proceedings of the Seventh Annual Symposium on Computational Geometry*, pages 281–288, North Conway, New Hampshire, July. Association for Computing Machinery.

Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco.

Nobuhiro Gō and Harold A. Scheraga. 1970. Ring closure and local conformational deformations of chain molecules. *Macromolecules*, 3(2):178–187, March–April.

N. Karmarkar and R. M. Karp. 1982. An efficient approximation scheme for the one-dimensional bin-packing problem. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 312–320, Los Angeles. IEEE Computer Society.

Charles Kittel. 1976. *Introduction to Solid State Physics*. John Wiley & Sons, New York, 5th edition.

Christopher Lee and S. Subbiah. 1991. Prediction of protein side-chain conformation by packing optimization. *Journal of Molecular Biology*, 217:373–388.

Cyrus Levinthal. 1968. Are there pathways for protein folding? *Journal de Chimie Physique*, 65(1):44–45, January.

Harry R. Lewis and Christos H. Papadimitriou. 1978. The efficiency of algorithms. *Scientific American*, 238(1):96–109, January.

Harry R. Lewis and Christos H. Papadimitriou. 1981. *Elements of the Theory of Computation*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

C. L. Liu. 1968. *Introduction to Combinatorial Mathematics*. McGraw-Hill, New York.

Joe Marks and Stuart Shieber. 1991. The computational complexity of cartographic label placement. Technical Report TR-05-91, Center for Research in Computing Technology, Harvard University, March.

Christos H. Papadimitriou and Kenneth Steiglitz. 1982. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, NJ 07632.

George N. Reeke, Jr. 1988. Protein folding: Computational approaches to an exponential-time problem. *Annual Reviews of Computer Science*, 3:59–84.

Edward M. Reingold, Jurg Nievergelt, and Narsingh Deo. 1977. *Combinatorial Algorithms: Thoery and Practice*. Prentice-Hall, Englewood Cliffs, New Jersey.

Andrzej Sikorski and Jeffrey Skolnick. 1990. Dynamic Monte Carlo simulations of globular protein folding/unfolding pathways. II. α-helical motifs. *Journal of Molecular Biology*, 212:819–836.

Andrew Streitwieser, Jr. and Clayton H. Heathcock. 1976. *Introduction to Organic Chemistry*. Macmillan, New York.

Ron Unger and John Moult. 1992. On the computational complexity of protein folding. Technical Report UMIACS-TR-92-15, Institute for Advanced Computer Studies, University of Maryland, January.

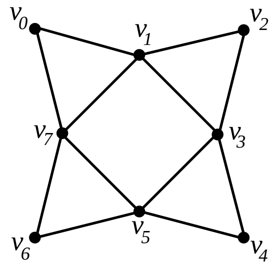

Figure 8: A sample graph $G = (V, E)$, where:

$$\begin{aligned} V &= (v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7) \\ E &= (\{v_0, v_1\}, \{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \\ &\quad \{v_4, v_5\}, \{v_5, v_6\}, \{v_6, v_7\}, \{v_7, v_0\}, \\ &\quad \{v_1, v_3\}, \{v_3, v_5\}, \{v_5, v_7\}, \{v_7, v_1\}) \end{aligned}$$

A Fallacious intractability arguments

Equating the notion of computational intractability with the size of a problem's solution space is fallacious. We illustrate this point first with two well-known problems from graph theory, and then with three problems in mathematical programming.

Hamiltonian circuits and Eulerian paths A graph $G = (V, E)$ consists of a set of vertices, V, and a set of edges, E. A sample graph is shown in Figure 8. A *path* π in G is a finite sequence $\langle v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(k)} \rangle$ of vertices from V, such that each consecutive pair of vertices is an edge in the graph. An *Eulerian path* in G is a path that traverses every edge in E exactly once, and a *Hamiltonian circuit* in G is a closed path that arrives at each vertex in V exactly once. The sequence $\langle v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_1, v_3, v_5, v_7, v_0 \rangle$ is an Eulerian path for the graph in Figure 8; the sequence $\langle v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_0 \rangle$ is a Hamiltonian circuit.

Finding Eulerian paths and Hamiltonian circuits seem at first to be very similar problems. In both cases the number of candidate solutions for each problem is exponential in the size of the graph. However, the problems have very different complexities. The Hamiltonian-circuit problem has been shown to be NP-complete, and therefore probably intractable (see Appendix B). By contrast, the problem of finding an Eulerian path can be solved efficiently: a simple polynomial-time algorithm is attributed to Euler (1707–1783) (Liu, 1968).

Linear programming and two variants Linear-programming (LP) problems arise frequently in operations research. Every LP problem can be stated in standard form as follows: Given the integer-valued (or rational-valued) matrix A and vectors b and c, with dimensions $m \times n$, m, and n, respectively ($m < n$), find a rational-valued vector x that minimizes the cost function $c \cdot x$ and satisfies the constraints $Ax = b$ and $x \geq \mathbf{0}$.

Integer linear programming (ILP) is very closely related to linear programming: integer linear programs have the same standard form as linear programs, except that the elements of

x are required to be integers. Zero-one linear programming (ZOLP) is even more restrictive: the elements of x must all be either 0 or 1.

The computational complexity of linear programming was an open problem for some time before a polynomial-time algorithm for LP was found (Papadimitriou and Steiglitz, 1982). However, even before the complexity question was resolved, large LP problems were routinely solved by Dantzig's simplex algorithm (Dantzig, 1963), which has a worst-case exponential-time complexity but is very efficient for most problems encountered in practice, even large ones involving hundreds or thousands of variables. Any ILP or ZOLP problem appears to be a restricted version of a corresponding LP problem that has infinitely more candidate x vectors. It is therefore tempting to conclude that ILP and ZOLP must be easier than LP—but this conclusion is erroneous, both in theory and in practice. ILP and ZOLP are both NP-complete, and only very small ILP and ZOLP problems can be solved exactly using current techniques (Papadimitriou and Steiglitz, 1982).

The two examples presented above show that detailed consideration of computational complexity is often necessary to understand the intrinsic difficulty of a problem, and that naive arguments based on the size of the solution space can sometimes be misleading.

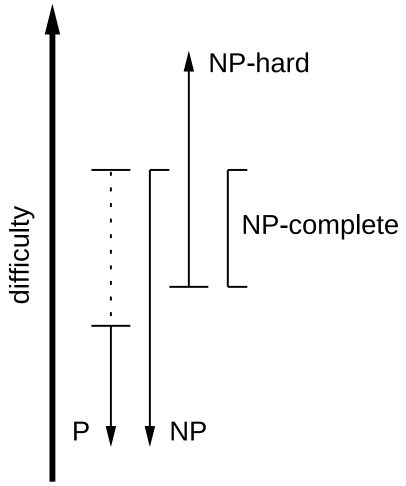

Figure 9: Diagram of some relationships between the computational classes P, NP, NP-complete, and NP-hard. Informally, the vertical axis represents the relative difficulty of the problems in each class. The class of NP-complete problems is the intersection of NP and the class of NP-hard problems. P is thought to be a proper subset of NP that excludes the NP-complete problems. The alternative possibility is that P=NP, as indicated by the broken line.

B Computational complexity theory

Overview The field of *computational complexity theory* concerns the efficiency of algorithms (Lewis and Papadimitriou, 1978, is a popular treatment). One branch of complexity theory addresses the space and time requirements of specific algorithms. For example, a straightforward analysis shows that Bubblesort, a sorting algorithm, runs in $O(n^2)$ time (*i.e.*, there is some constant, c, such that the running time of Bubblesort is bounded above by cn^2 for all $n \geq 0$, where n is the number of items to be sorted). A more ambitious goal of complexity theory is to determine the intrinsic difficulty of certain problems, *i.e.*, to prove something about the complexity of the best possible algorithm for a particular problem. For example, a simple argument shows that no comparison-based sorting algorithm can require fewer than than $O(n \log n)$ comparisons (Aho et al., 1974).

The theory of NP-completeness is in the latter vein; it concerns the worst-case complexity, as opposed to average-case or best-case complexity, of a particular class of problems. A problem is NP-hard if the first of the following conditions holds, or NP-complete if both can be demonstrated. (The class of NP-complete problems is a subset of the class of NP-hard problems. This and other relationships are diagrammed in Figure 9.)

1. A known NP-complete problem is *polynomial-time reducible* (or *efficiently reducible*) to the problem in question. In other words, there exists an algorithm for transforming any instance of a known NP-complete problem into some instance of the candidate problem, such that

   (a) a solution to the latter leads directly to a solution to the former,

(b) the transformation runs in time polynomial in the size of its input, and

(c) the generated problem instance occupies space polynomial in the size of the input (this is automatically implied by (b)).

2. The problem is in the class NP, which means that it could be solved in polynomial time on a nondeterministic Turing machine (NDTM). The essence of this requirement is *efficient verifiability*: if the problem has solutions, then at least one of them can be verified in polynomial time. (The computational effort required to generate candidate solutions and to identify incorrect ones is ignored. Thus, an NDTM can be thought of informally as a computer that can examine all potential solutions to a problem simultaneously. The notion of an NDTM is only an abstract model of computation, not a design for a realistic computing device.)

These tests are such that *a polynomial-time solution to any one NP-hard problem would lead directly to polynomial-time solutions for all NP-complete problems*. This is a consequence of Cook's Theorem (Cook, 1971)—the non-intuitive result that all problems in NP are reducible to SATISFIABILITY, by definition the first known NP-complete problem. From Cook's Theorem and the tests outlined above, it follows that any two NP-complete problems are reducible to each other. Therefore the NP-complete problems form an equivalence class which lies either entirely within, or entirely outside, P, the class of problems with polynomial-time worst-case complexity. Furthermore, if P contains the NP-complete problems, then it must be equivalent to NP.

It has not been proven that NP-complete problems are not in P. However, given the large number of problems that have been shown to be NP-complete (some of the better-known problems are listed in Figure 10), none of which are known to be solvable in polynomial time, it is extremely unlikely that NP-complete problems can be solved efficiently. Thus, showing that a problem is NP-hard is a very cogent argument—usually the most rigorous argument available—that it lies in "the abyss of inherent intractability" (Garey and Johnson, 1979, page ix).

Limitations The principle of reducibility, which is fundamental to NP-completeness theory, is based on thought experiments about *exact* algorithms and guarantees on their performance for *all* possible inputs. We explain two consequent limitations.

Firstly, an NP-hardness result describes only worst-case behavior. For any NP-hard problem there may be some subset of problem instances, *i.e.*, a *restricted* form of the problem, for which the existence of an efficient algorithm would not imply the equivalence of P and NP. For example, CLIQUE is an NP-complete problem for general graphs, but the restricted version of the problem which specifies that the given graph is planar (capable of being drawn without edge crossings) can be solved in polynomial time (Garey and Johnson, 1979). Thus, the best-case complexity of the general CLIQUE problem is polynomial: a best-case polynomial-time algorithm for CLIQUE might first check whether the graph is planar (this can be done in polynomial time (Reingold et al., 1977)), and then execute either the efficient algorithm for planar graphs or an exponential-time algorithm for non-planar graphs. Likewise, the theory of NP-completeness cannot be applied directly to average-case complexity analysis. Any such analysis depends greatly on the assumed distribution of problem instances. There is no body of theory similar to NP-completeness for average- or best-case complexity, though these measures are often of practical significance.

SATISFIABILITY
INSTANCE: A set of variables, V, and a Boolean expression E over V. (Any of the 16 possible logical connectives are allowed in E, though the problem remains NP-complete even if only the connectives $\wedge, \vee, \rightarrow, \neg$ are allowed.)
QUESTION: Is there a truth assignment for the variables in V that satisfies E?

TRAVELING SALESMAN
INSTANCE: A set C of m cities, a distance $d(c_i, c_j) \in Z^+$ for each pair of cities $c_i, c_j \in C$, and a positive integer B. (Z^+ is the set of positive integers.)
QUESTION: Is there a tour of C having length B or less, *i.e.*, a permutation $\langle c_{\pi(1)}, c_{\pi(2)}, \ldots, c_{\pi(m)} \rangle$ of C such that $[\sum_{i=1}^{m-1} d(c_{\pi(i)}, c_{\pi(i+1)})] + d(c_{\pi(m)}, c_{\pi(1)}) \leq B$?

BIN PACKING
INSTANCE: A finite set U of items, a size $s(u) \in Z^+$ for each $u \in U$, a positive integer bin capacity B, and a positive integer K.
QUESTION: Is there a partition of U into disjoint sets ("bins") $U_1, U_2, \ldots, U_K$ such that the sum of the sizes of the items in each U_i is B or less?

CLIQUE
INSTANCE: A graph $G = (V, E)$, and a positive integer $K \leq |V|$.
QUESTION: Is there a clique of at least size K in G? (A *clique* is a subset $V' \subseteq V$ such that every pair of vertices in V' is joined by an edge in E.)

Figure 10: Some NP-complete problems.

Secondly, an NP-hard optimization problem might yield to solution by an approximation algorithm. An approximation algorithm (or ϵ-approximation algorithm) is one that will always find a solution that is within a multiplicative factor ϵ of optimal in polynomial time. For some NP-hard optimization problems, it is true that no efficient ϵ-approximation algorithm can exist if P$\neq$NP. For example, the existence of an ϵ-approximation algorithm for the optimization version of TRAVELING SALESMAN, regardless of what ϵ is for that algorithm, would imply the equivalence of P and NP (Papadimitriou and Steiglitz, 1982). However, for other NP-hard optimization problems, approximation algorithms with known performance guarantees have been found. The First Fit Decreasing algorithm for BIN PACKING, for example, is straightforward and efficient, and is guaranteed to produce solutions with no more than $\frac{11}{9}$OPT(I) + 4 bins, where OPT(I) is the minimum number of bins for problem instance I (Garey and Johnson, 1979); an approximation algorithm with an even better bound of OPT$(I) + O(\log^2$ OPT$(I))$ is also known (Karmarkar and Karp, 1982). Thus, while it is true that no NP-hard optimization problem can be solved exactly in polynomial time unless P=NP, some of these problems respond better to approximation and heuristic techniques than others.