# On the Computational Complexity of Sequence Design Problems (Extended Abstract)

William E. Hart*

Sandia National Laboratories
Algorithms and Discrete Mathematics Department
P. O. Box 5800
Albuquerque, NM 87185-1110
August 9, 1996

**Abstract**

Inverse protein folding concerns the identification of an amino acid sequence that folds to a given structure. Sequence design problems attempt to avoid the apparent difficulty of inverse protein folding by defining an energy that can be minimized to find protein-like sequences. We evaluate the practical relevance of two sequence design problems by analyzing their computational complexity. We show that the canonical method of sequence design is intractable, and describe approximation algorithms for this problem. We also describe an efficient algorithm that exactly solves the grand canonical method. Our analysis shows how sequence design problems can fail to reduce the difficulty of the inverse protein folding problem, and highlights the need to analyze these problems to evaluate their practical relevance.

## 1 Introduction

The goal of the inverse protein folding problem (IPF) is to design a polymer sequence that folds to a given target conformation. Three criteria have been proposed for evaluating the success of a protein sequence that has been designed for a target conformation [1, 7]. First, the protein sequence should fold to the target conformation. This means that the energy of the sequence in the target conformation is not greater than the energy of the sequence in any other conformation. Second, the target conformation is the only conformation in which the sequence folds to the minimal energy. This means that there is no *degeneracy* of ground states for the sequence. Yue and Dill [7] weaken this criterion to require that the degeneracy of the sequence be no greater than the degeneracy of any other sequence that folds to the target conformation. Third, there should be a large gap in the energy of the sequence in the target conformation and the energy of the sequence in any other conformation.

At present very little is known about the computational complexity of IPF. IPF appears to involve a search over sequences as well as a search over conformations to guarantee that the sequence has minimal degeneracy. No algorithm is known that can reliably solve IPF without this exhaustive search, which involves an exponential number of conformations. In fact, we conjecture that IPF is intractable (i.e., NP-hard).[1]

*wehart@cs.sandia.gov; http://www.cs.sandia.gov/~wehart/

[1] A brief description of computational intractability and its relationship to NP-hardness is given in the appendix.

# DISCLAIMER

**Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.**

# DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Recently, a variety of methods have been described that attempt to solve IPF without performing this exhaustive search [1, 3, 4, 5, 6, 7]. These methods are heuristic algorithms because they do not guarantee that the sequence engineered by the algorithm solves IPF. In general terms, these methods attempt to capture two aspects of IPF that are intuitively related to the criteria described above: (i) positive design – the sequence folds to the given conformation, and (ii) negative design – the sequence does not fold to other structures with the same or lower energy. These algorithms have been tested on small conformations for which exhaustive search of the exact solution is possible, as well as conformations taken from the PDB database.

These heuristic methods can be separated into two categories. First, the authors use observations about the properties of proteins to justify algorithms that design sequences [3, 7]. These algorithms are heuristics that run quickly but are not guaranteed to solve IPF. The second category of heuristic methods are those in which the authors propose an alternative formulation of IPF [1, 4, 5, 6]. This alternative formulation attempts to capture the positive and negative design issues by defining a heuristic sequence design (HSD) problem. An implicit assumption of this approach is that a sequence that satisfies the HSD problem is likely to solve IPF.

Ideally, the HSD problem should not require the exhaustive enumeration that is currently used to exactly solve IPF. Thus it should be possible to find the sequence that satisfies the HSD problem for the target conformation in a polynomial number of steps (in the length of the protein sequence). If this is not true, then the reformulation of IPF is less interesting because it does not reduce IPF to a problem that can be solved efficiently. Although analyses of intractable HSD problems may provide insight into IPF, only problems that can be solved efficiently are of practical relevance.

In this paper, we evaluate the practical relevance of two HSD problems by examining their computational complexity. The problems that we analyze are the 'canonical method' of Shakhnovich and Gutin [5] and the 'grand canonical method' of Sun *et al.* [6]. To solve these HSD problems, these authors use stochastic search algorithms that provide only weak guarantees that the best sequence is generated. Consequently, the computational complexity of these two HSD problems remains an open question.

Our results show that the canonical method is intractable (i.e. NP-hard), but we describe an algorithm that efficiently constructs sequences that approximate the best canonical sequence. Surprisingly, for the 2D cubic lattice it is possible to efficiently construct a sequence whose energy (as defined by the canonical method) is no greater than one above the energy of the best sequence. For the 3D cubic lattice we show that the algorithm efficiently constructs a sequence whose energy is guaranteed to be within a factor of two of the energy of the best sequence. For the grand canonical method we describe a polynomial time algorithm that constructs sequences whose energy is optimal for 2D and 3D cubic lattices.

## 2 Definitions

In this paper we consider two HSD problems defined for the HP lattice model []. This model uses uses contact energies to determine the energy of a protein sequence in the target conformation. The HP model categorizes amino acids as either hydrophobic (nonpolar) or hydrophilic (polar). The contact energy gives an energy of -1 to hydrophobic-hydrophobic contacts and an energy of 0 to all other contacts.

Let a target conformation be described by a graph $G = (V, E)$ with vertices $V$, that correspond to amino acids, and edges $E \subseteq V\text{x}V$, that define a self-avoiding walk on a 2D or 3D cubic lattice. Recall that $|V|$ is the number of vertices in $V$. Let $\mathcal{G}_k$ be the set of all target conformations for which the length of the chain is $k$ (i.e., $|V| = k$). Given a conformation $G = (V, E)$, we can construct a *contact graph* $\bar{G} = (V, \bar{E})$ induced by $G$, where an edge $(a, b)$ is in $\bar{E}$ if $a, b \in V$

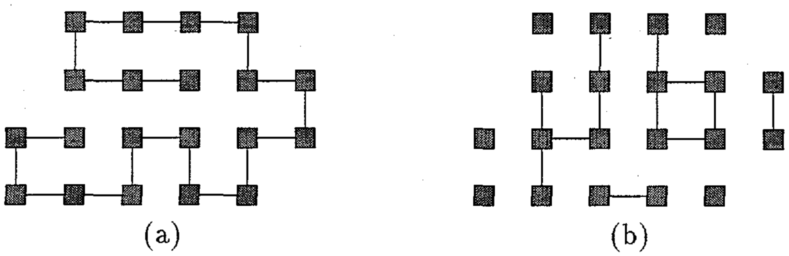


Figure 1: Illustration of (a) a target conformation along with (b) its corresponding contact graph.

and $(a, b)$ is an edge in the lattice. Figure 1 shows a target conformation in a 2D cubic lattice along with its corresponding contact graph. For a sequence $s$, let $E(G, s)$ be the conformational energy of $s$ when the vertices of $G$ are labeled with the sequence of amino acids defined by $s$; the calculation of $E(G, s)$ implicitly depends upon the energy matrix defined by the lattice model. Let $\mathcal{S}_G$ be the set of sequences that achieve their lowest energy in the target conformation $G$. Formally, $\mathcal{S}_G = \{s \in \{H, P\}^n \mid E(G, s) \leq E(G', s), \forall G' \in \mathcal{G}_n\}$, where $n = |V|$. We say that a sequence $s$ *folds* to $G$ if $s \in \mathcal{S}_G$.

## 3 The Canonical Method

Shahknovich and Gutin [5] define the canonical method, a HSD problem for protein sequences in the HP model. Shahknovich and Gutin observe that for any target conformation, the conformational energy can be minimized simply by using the sequence of all hydrophobics, but that this sequence is unlikely to achieve its lowest energy with the given target conformation. To account for this, they limit the number of hydrophobics that can be used in a protein sequence by fixing the ratio between hydrophobic and hydrophilic amino acids.

We formulate the HSD problem posed by Shaknovich and Gutin as follows. Let $L$ be a 2D or 3D cubic lattice and let $G = (V, E)$ be a target conformation in $L$. Let $\lambda \in \mathbf{Q}$ represent the fraction of hydrophobics that we will allow in a protein sequence. The *canonical method* is the problem of minimizing $E(G, s)$ subject to the constraint that no more that $\lceil \lambda n \rceil$ hydrophobics be used to design $s$. Shahknovich and Gutin [5] use a Monte Carlo method to search through the space of conformations to find a sequence that minimizes $E(G, s)$ subject to this constraint. This type of stochastic search algorithm only provides a weak probabilistic guarantee that the optimal sequence is generated.

In the following section, we consider the possible intractability of this problem. We prove that this problem is intractable by showing that it is NP-hard for both the 2D and 3D cubic lattices. Then we present positive results that show that this problem is approximable in polynomial time. For the 2D lattice, we show that a sequence can be designed in polynomial time whose energy differs from the energy of the best sequence by at most one. For the 3D lattice, we show that a sequence can be found in polynomial time whose energy is within a factor of 2 of the energy of the best sequence.

### 3.1 Intractability Results

Since the canonical method is an optimization problem, we define the following decision problem:

**$(L, \lambda, HP)$-Inverse Protein Folding**
**Given:** A conformation $G = (V, E)$ embedded in $L$; an integer $K$
**Question:** Is there a protein sequence $s \in \{H, P\}^n$, $n = |V|$, with $\lceil \lambda n \rceil$ or fewer hydrophobic amino acids such that $E(G, s) \leq K$?

The following theorem shows that $(L, \lambda, HP)$-IPF is NP-complete for the 2D and 3D cubic lattices. This shows that it is highly unlikely that there exists a polynomial time algorithm that exactly solves this problem for all instances of $(L, \lambda, HP)$-IPF. Note that this is a *robust* intractability argument that proves NP-completeness for an infinite class of problems that are indexed by the value $\lambda$. Thus for any value of $\lambda$, the problem remains difficult.

**Theorem 1** Let L be either the 2D or 3D cubic lattice. Then $(L, \lambda, HP)$-IPF is NP-complete.

This intractability argument could be strengthened in two ways. It would be interesting to determine whether this problem remained intractable for other lattice models using contact potentials with different alphabets and contact energy matrices. In fact, we suspect that this problem remains NP-complete for a wide variety of other lattice models.

This argument would be also be strengthened if we could restrict the target conformations to the "compact" and "native-like" structures to which IPF is likely to be applied. The proof of Theorem 1 uses sparse, elongated target conformations to prove that $(L, \lambda, HP)$-IPF is NP-complete. It is unclear whether this result would hold if $(L, \lambda, HP)$-IPF was restricted to more interesting target conformations. One possible restriction is the definition of compact conformations used by Deutsch and Kurosky [1]. For a sequence of length $n$, they call a conformation compact if it has $(n-2)/2$ or more contacts. However, the proof of Theorem 1 uses target conformations that are compact by this definition.

## 3.2 Approximation Algorithms

Performance guaranteed approximation algorithms for $(L, \lambda, HP)$-IPF quickly design a sequence that is guaranteed to have an energy that is close to the energy of the best possible sequence. Because a two-dimensional target conformation can be embedded on a 3D lattice, the set of possible contact graphs for the 2D cubic lattice is a subset of the set of possible contact graphs for the 3D cubic lattice. Consequently, an approximation algorithm for the 3D lattice will be an approximation algorithm for the 2D lattice. In this section we describe a performance guaranteed approximation algorithm for the 3D lattice and refine the analysis of this algorithm to prove a much tighter performance guarantee on the 2D lattice.

Figure 2 describes Algorithm A. Algorithm A labels components and parts of components in the contact graph with $\lceil \lambda n \rceil$ hydrophobics. This algorithm divides connected components of the contact graph into three classes: (1) components with no cycles, (2) components with one cycle and (3) components with two or more cycles. Given this division, Algorithm A attempts to label the amino acids in components with the most cycles first, which maximizes the total energy of protein sequence.

Figure 3 illustrates the application of Algorithm A for different values of $\lambda$; black squares represent hydrophobic amino acids and white squares represent hydrophilic amino acids. The initial classification of connected components in step 2 can be done efficiently using a variation of depth first search, and all other steps clearly have polynomial complexity. Consequently, the complexity of Algorithm A is polynomial in the number of vertices in the target conformation.

Let $A(G, \lambda)$ be the energy of the sequence constructed by Algorithm A on target conformation $G$ for a given value of $\lambda$. Let $OPT(G, \lambda)$ be the energy of the best sequence possible. The following

```
1. Compute the contact graph $\bar{G}$ from $G$. Let $J = \lceil \lambda n \rceil$.

2. Classify the connected components in $\bar{G}$ into three classes: (1) components with no cycles,
      (2) components with one cycle, and (3) components with two or more cycles.
      Let $n_i$ equal the total number of vertices in the components in class (i).

3. If $J \geq n_2 + n_3$, then
      Label all components in classes (2) and (3) as hydrophobic,
      Sort the components in class (1) by size
      Iteratively label the components in class (1) as hydrophobic from smallest to largest
      Partially label the last component as hydrophobic using a DFS method, and
      Label remaining unlabeled vertices as hydrophilic

4. If $n_2 + n_3 > J \geq n_3$, then
      Label all components in class (1) as hydrophilic,
      Label all components in class (3) as hydrophobic,
      Iteratively label the components in class (2) as hydrophobic in any order
      Partially label the last component as hydrophobic using a DFS method such
        that the cycle is filled first (if possible), and
      Label remaining unlabeled vertices as hydrophilic

5. If $n_3 > J$, then
      Label all components in class (1) as hydrophilic,
      Iteratively label the components in class (3) as hydrophobic in any order
      Partially label the last component as hydrophobic using a DFS method such
        that the cycle is filled first (if possible), and
      Label remaining unlabeled vertices as hydrophilic
```

Figure 2: Algorithm A

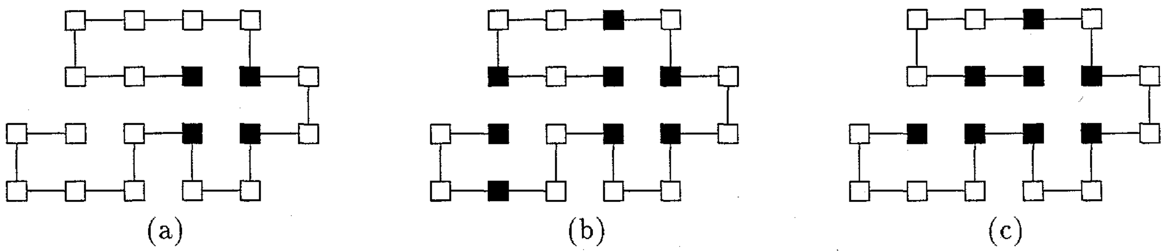


Figure 3: Illustration of the application of Algorithm A to the target conformation shown in Figure 1a: (a) $\lambda = 1/5$, (b) $\lambda = 2/5$, and (c) $\lambda = 2/5$. The sequence shown in (c) has the same energy as the sequence shown in (b), but it has fewer exposed hydrophobics. Algorithm A arbitrarily chooses one of these two solutions.

proposition proves that Algorithm A generates a sequence whose energy is within a multiplicative factor of two of the optimal energy.

**Proposition 1** $A(G,\lambda) \leq \frac{1}{2}OPT(G,\lambda)$.

**Proof.** Let $E_i^k$ equal the energy of a connected component of the contact graph from class $i$ with $k$ vertices. Note that $E_1^k = -k+1$, $E_2^k = -k$ and $E_3^k \leq -k-1$. Consequently the optimal sequence labels components hydrophobic in class $i+1$ before labeling components hydrophobic in class $i$.

If Algorithm A executes step 3, then $A(G,\lambda) = OPT(G,\lambda)$ because this step minimizes the number of components included from class (1), thereby maximizing the total energy of the designed sequence.

Otherwise, a sequence $s$ has been designed such that $E(G,s) \leq -J+1$. For the optimal sequence each hydrophobic can have at most four contacts except for two hydrophobics that can have five. Also, at least eight of the vertices with degree three or less, because there are eight "corners" to the three-dimensional conformation. Thus we have

$$OPT(G,\lambda) \geq \begin{cases} -3J/2 & ,J \leq 8 \\ -(24+5)/2 & ,J = 9 \\ -(24+10)/2 & ,J = 10 \\ -(24+10+4(J-10))/2 = -2J+3 & ,J > 10 \end{cases}$$

Consequently, $A(G,\lambda) \leq \frac{1}{2}OPT(G,\lambda)$. ■

Proposition 1 shows that there exists a performance guaranteed approximation algorithm for the canonical method for both the 2D and 3D cubic lattice. On the 2D lattice, there are only nine topological classes of contact graphs possible because there are at most two vertices in the contact graph that have degree three (these are the endpoints of the protein chain). The following lemma proves that on a 2D cubic lattice Algorithm A generates a sequence whose energy comes within one energy unit of the optimal energy.

**Proposition 2** $A(G,\lambda) \leq OPT(G,\lambda)+1$.

**Proof.** Recall that the optimal sequence labels components hydrophobic in class $i+1$ before labeling components hydrophobic in class $i$.

If Algorithm A executes step 3, then $A(G,\lambda) = OPT(G,\lambda)$ because this step minimizes the number of components included from class (1), thereby maximizing the total energy of the designed sequence.

If Algorithm A executes step 4, then $A(G,\lambda) \leq OPT(G,\lambda)+1$. Because each component in class (2) with $k$ vertices adds $-k$ to the energy, if any collection of components from this class can be filled exactly the total energy of these components will be $-(J-n_3)$. If any component is only partially filled with $k$ vertices, then its energy is at most $-k+1$. Since the optimal labeling may exactly fill a set of components from class (2), this implies that $A(G,\lambda) \leq OPT(G,\lambda)+1$.

If Algorithm A executes step 5, then $A(G,\lambda) \leq OPT(G,\lambda)+1$. On the 2D cubic lattice there exists two topologically distinct possible connected components in class (3). Furthermore, there can exist at most a single component from these two categories because there are at most two vertices with degree three in the contact graph. Now if this component were completely filled, it would have energy $-J-1$. Since it cannot, the optimal energy is greater than or equal to $-J$. Algorithm A fills in this component to form a connected subcomponent of hydrophobics that has energy no greater than $-J+1$. Consequently, $A(G,\lambda) \leq OPT(G,\lambda)+1$. ■

## 4 The Grand Canonical Method

Sun *et al* [6] define the grand canonical method, which is a HSD problem for a variation of the HP model. Like the HP model, their model uses a contact energy potential that categorizes amino acids as either hydrophobic (nonpolar) or hydrophilic (polar). The contact energy gives an energy of -2 to hydrophobic-hydrophobic contacts, an energy 1 for every solvent accessible site on a hydrophobic amino acid, and 0 for all other interactions. This contact potential is used to eliminate the need to specify the fraction of hydrophobics in the sequence needed by the canonical method. Because hydrophobics are penalized for their exposure to solvent, this contact potential implicitly limits the number of hydrophobics in the sequence.

The goal of the grand canonical method is to label a subset of the vertices in a contact graph hydrophobic such that $E(G,s)$ is minimal. The remainder of this section shows that this problem can be solved exactly in polynomial time for contact graphs embedded on 2D and 3D cubic lattices. Let $c(v_i, V)$ be the number of contacts that the $i$th amino acid makes with the amino acids in $V$, and let $w(v_i)$ be the number of solvent-residue contacts that the $i$th amino acid makes. Now suppose there exists an amino acid $v_i$ for which $w(v_i) - 2c(v_i, V_H) > 0$, where $V_H \subseteq V$ are the amino acids labeled hydrophobic. If this amino acid is labeled a hydrophilic, then the energy of the conformation will decrease.

This observation leads to a simple greedy algorithm that iteratively scans each of the amino acids $v_i$, relabeling them as hydrophilic if $w(v_i) - 2c(v_i, V_H) \geq 0$ and removing $v_i$ from $V_H$. Figure 4 describes this algorithm, Algorithm B. The DOLABEL subroutine implements this greedy relabeling of hydrophobic amino acids to hydrophilic amino acids. The MAIN routine calls DOLABEL to find an initial set of hydrophobic amino acids. It then performs several additional calls to DOLABEL to see whether a lower energy sequence could be designed by labeling the endpoints ($v_1$ and $v_n$) of the sequence hydrophilic.

Proposition 3 proves that the sequence $s$ designed by Algorithm B is the sequence that minimizes $E(G,s)$. Subroutine DOLABEL makes at most $n$ passes through the outer loop, since each pass ensures that at least one amino acid is labeled hydrophilic. The inner loop requires a check of at most $n$ amino acids to pick the amino acid $v$ for which $w(v) - 2c(v, V_H)$ is maximized. Thus, the complexity of Algorithm B is $O(n^2)$.

**Proposition 3** Let $s$ be the sequence generated by Algorithm B for a contact graph $G$ embedded on the 2D or 3D cubic lattice. Then $s$ minimizes $E(G,s)$.

**Proof.** We consider the performance of Algorithm $B$ for contact graphs embedded on the 3D cubic lattice. Since the 2D cubic lattice is a subset of the 2D cubic lattice, it follows that this analysis also applies to Algorithm $B$ when restricted to contact graphs embedded on the 2D cubic lattice.

We begin by showing that the optimal set of vertices labeled hydrophobic is $V^* \subseteq \bar{V}_1$. Suppose that $V^* \not\subseteq \bar{V}_1$. Consider the sequence of vertices that are labeled hydrophilic and removed from $V_t$ by Algorithm $B$. Let $v$ be the first vertex in this sequence that is contained in $V^*$, and let $V_t$ be the set of hydrophobic vertices just before $v$ is labeled hydrophilic. Since $v \notin \bar{V}_1$ we know that $w(v) - 2c(v, V_t) \geq 0$. We also know that $w(v) - 2c(v, \bar{V}_1 < 0$ because $v \in V^*$. This implies that $c(v, V^*) > c(v, V_t)$, but this is impossible because $V^* \subseteq V_t$. Consequently, $V^* \subseteq \bar{V}_1$.

Suppose that $G$ is embedded on the 3D cubic lattice. Then $w(v)+c(v, V_H) \leq w(v)+c(v, V) \leq 5$. For $v \in \bar{V}_1$, $w(v) - 2c(v, \bar{V}_1) < 0$. Consequently, every $v \in \bar{V}_1$ has the property that it is one of the following $(c(v), w(v))$ pairs: (#,0), (4,1), (3,2), (3,1), (2,3), (2,2), (2,1), or (1,1). Among these pairs, only the (2,3) pair has the property that $w(v) - c(v) > 0$.

```
MAIN(V)
    BEGIN
    $(e_1, \bar{V}_1) = $ DOLABEL($V$)
    $(e_2, \bar{V}_2) = $ DOLABEL($\bar{V}_1 - \{v_1\}$)
    $(e_3, \bar{V}_3) = $ DOLABEL($\bar{V}_1 - \{v_2\}$)
    $(e_4, \bar{V}_4) = $ DOLABEL($\bar{V}_1 - \{v_1, v_2\}$)

    $i = \arg\min_j e_j$
    for all $v \in \bar{V}_i$ label $v$ hydrophobic
    for all $v \in V - \bar{V}_i$ label $v$ hydrophilic
    END


$(e, V') = $ DOLABEL($\bar{V}$)
    BEGIN
    $V_1 = \bar{V}$
    $t = 1$
    DO
        Pick $v' \in V_t$ such that $w(v') - 2c(v', V_t) \geq w(v) - 2c(v, V_t)$ for all $v \in V_t$
        IF $(w(v') - 2c(v', V_t) \geq 0)$
            $V_{t+1} = V_t - \{v'\}$
        ENDIF
        t++
    UNTIL $(|V_t| = 0$ OR $|V_t| = |V_{t-1}|)$

    $V' = V_t$
    $e = \sum_{v \in V'} (w(v) - c(v, V'))$
    END
```

Figure 4: Algorithm B.

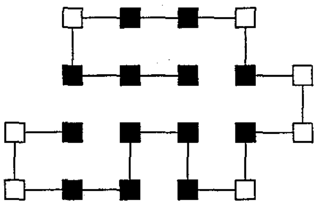


Figure 5: Illustration of the application of Algorithm B to the 2D target conformation in Figure 1a.

If $\bar{V}_1$ is composed of vertices that are non-(2,3) pairs, then it is optimal. Removing any vertex can only increase the total energy. Consequently, we must focus on removing the vertices that are (2,3) pairs. There can only be two such vertices, because these vertices must be the endpoints of the amino acid. Consequently, we can exhaustively determine whether labeling these vertices hydrophilic leads to a decrease in the total energy. This is reason Algorithm *B* performs the three additional calls to DOLABEL. It follows that the optimal set of hydrophobics is identified by Algorithm *B*. ■

Figure 5 illustrates the application of Algorithm B on to the target structure shown in Figure 1a. Algorithm B can only label solvent accessible vertices as hydrophilics, so this HSD problem preserves a reasonable notion of a hydrophobic core.

## 5 Discussion

Our analysis of the canonical and grand canonical methods illustrates two ways in which computational complexity can provide insight into sequence design. First, a complexity analysis provides a well defined measure of the practical relevance of a HSD problem. Our intractability analysis of the canonical method shows how sequence design problems can fail to reduce the apparent difficulty of the inverse protein folding problem. Because this problem is intractable, its practical utility seems quite limited. Our analysis of the grand canonical method demonstrates that sequences can be efficiently designed, thereby ensuring that this method can be used in practical contexts. Although careful experimentation is also necessary to evaluate the practical utility of HSD problems, computational analyses provide a rigorous basis for evaluating their practical utility. Prior work with both of these HSD problems used weak stochastic methods to design sequences, so our analyses provide the first critical evaluation of the computational difficulty of these problems.

The second way complexity analyses can provide insight is through the rigorous evaluation of the HSD problems. Our analysis of approximation algorithms for the canonical method suggests that it is relatively easy to find near optimal sequences for the 2D cubic lattice, since it is possible to quickly determine sequences that differ from the optimal by at most one. However, our weaker bound on the 3D cubic lattice might indicate that finding near optimal sequences for this problem is more difficult.

Similarly, our analysis has led to new understandings about the relative strengths and weaknesses of these models. For example, Figure 3 illustrates how the canonical method can be indifferent to factors like the number of solvent accessible hydrophobic amino acids. Also, a careful examination of the grand canonical method reveals that there can exist subsets of $V^*$ that can be labeled either hydrophilic or hydrophobic (as a whole) without affecting the total energy of the sequence. This implies that there may not be a single best sequence predicted by the grand canonical method, but a potentially large number of best sequences. This observation suggests that we should be careful when evaluating the solution to this method in the context of the IPF problem.

This work has raised a variety of open problems related to IPF. The complexity of IPF remains the most important unresolved question, and we conjecture that solving IPF is in fact NP-hard. As we mentioned earlier, it would be interesting to evaluate the complexity of the canonical method when the space of possible conformations is restricted. Our analysis of the grand canonical method leaves several questions unanswered. First, can this analysis be extended to related models where the contact energy is $-\alpha$, $\alpha \in \mathbf{Q}^+$? For specific values of $\alpha$ this problem can be solved efficiently, but the analysis presented in here does not generalize to certain cases. Similarly, it would be interesting to extend the analysis of the grand canonical method to handle contact graphs generated by off-lattice conformations.

## Acknowledgements

This work was supported by the Applied Mathematical Sciences program, U.S. Department of Energy, Office of Energy Research, and was performed at Sandia National Laboratories, operated for the U.S. Department of Energy under contract No. DE-AC04-94AL85000. I am grateful to Sorin Istrail and Sarina Bromberg for their critical feedback. I also thank Johnathan Atkins and to the members of the SNL discrete algorithms group for their helpful discussions.

## Appendix

Computational intractability refers to our inability to construct efficient (i.e., polynomial time) algorithms that can solve a given problem. Here, "inability" refers to both the present state-of-the-art of algorithmic research as well as the possible mathematical statement that no such algorithms exist. Customary statements about the intractability of a problem are made by showing that the problem is NP-complete. The best known algorithm for any NP-complete problem takes more than a polynomial number of computational steps, which makes these problems "practically intractable."

The class of problems NP includes a wide variety of notoriously difficult combinatorial problems, such as the traveling salesman problem, various scheduling problems, and network design. An *instance* is the information needed to specify a particular example that needs to be solved (e.g., the target conformation). Problems in NP have the property that given an instance of the problem and a potential solution, one can efficiently determine whether the potential solution actually solves the problem instance. A problem is NP-complete if (a) it belongs to NP, and (b) if there is a polynomial algorithm that can solve this problem then this algorithm can be adapted to solve all of the other problems in NP. Hence, the problem is at least as hard as every other problem in NP. For a thorough treatment of NP-completeness see Garey and Johnson [2].

Formally, NP-complete problems are decision problems, for which the answer is either yes or no. Optimization problems are not directly considered within the framework of NP-completeness. However, optimization problems can be transformed into a decision problem by introducing a threshold $B$ and asking whether a solution with value less than or equal to $B$ exists. The corresponding optimization problem is at least as hard as the decision problem, since finding the optimal solution would answer this decision problem for every value of $B$. Consequently, an optimization problem is NP-hard if its corresponding decision problem is shown to be NP-complete.

## References

[1] J. M. Deutsch and T. Kurosky, *New algorithm for protein design*, Physical Review Letters, 76 (1996), pp. 323–326.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability - A guide to the theory of NP-completeness*, W.H. Freeman and Co., 1979.

[3] S. Kamtekar, J. M. Schiffer, H. Xiong, J. M. Babik, and M. H. Hecht, Science, 262 (1993), pp. 1680–1685.

[4] T. Kurosky and J. M. Deutsch, *Design of copolymeric materials*, J. Phys. A, 27 (1995), pp. L387–L393.

[5] E. I. Shakhnovich and A. M. Gutin, *Engineering of stable and fast-folding sequences of model proteins*, Proc. Natl. Acad. Sci., 90 (1993), pp. 7195–7199.

[6] S. Sun, R. Brem, H. S. Chan, and K. A. Dill, *Designing amino acid sequences to fold with good hydrophobic cores*, Protein Engineering, 9 (1996). (in press).

[7] K. Yue and K. A. Dill, *Inverse protein folding problem: Designing polymer sequences*, Proc. Natl. Acad. Sci. USA, 89 (1992), pp. 4163–7.

# Technical Appendix to "On the Computational Complexity of Sequence Design Problems"

William Hart
wehart@cs.sandia.gov

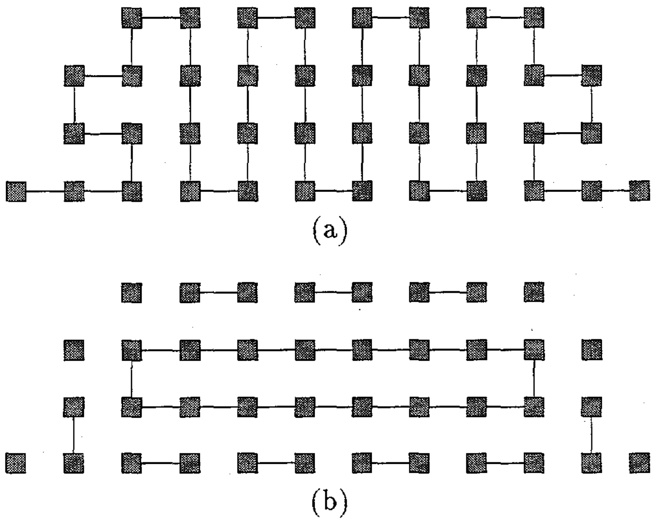
## A Proof of Theorem 1

To prove Theorem 1, we use a reduction from the following NP-complete problem [2].:

SUBSET SUM
**Given:** $A = \{a_1, \ldots, a_k\}$, $a_i \in \mathbf{Z}^+$; $B \in \mathbf{Z}^+$
**Question:** Does there exist a subset $A' \subseteq A$ such that

$$\sum_{a \in A'} = B?$$

**Proof.** The following reduction is used to transform an instance of SUBSET SUM to an instance of $(L, \lambda, HP)$-IPF. Let $\nu, \delta \in \mathbf{Z}^+$ such that $\lambda = \nu/\delta$. Let $A = \{a_1, \ldots, a_k\}$. For each $a_i$ we construct a subconformation of the final conformation as shown in Figure 6a. The subconformations for $a_i$ and $a_{i+1}$ are connected together by sharing the points at the ends of the chain folded in Figure 6a. Thus the complete conformation, $G$, can be viewed as a chain of independent subconformations; this conformation has $n = 1 + 7k + \sum_{a \in A} 8a$ vertices. The transformation constructs this chain, and defines $K = -4B$ and $\lambda = 4B/n$.


(a)

(b)

Figure 6: Illustration of the subconformation used to construct contact cycles of length $4a$ for every $a \in A$: (a) the subconformation, and (b) the corresponding contact graph.

Suppose there a subset $A' \subseteq A$ such that $\sum_{a \in A'} a = B$. Then we can design a protein sequence, $s$, for this conformation by labeling the subconformations corresponding to the $a \in A'$ as follows. Label each of the amino acids that has zero or one contacts as a hydrophilic and label remaining amino acids as hydrophobics. For subconformations corresponding to the $\bar{a} \in A - A'$, label all amino acids as hydrophilics. For each $a \in A'$, the corresponding subconformation has $4a$ hydrophobic-hydrophobic contacts, and all other subconformations have none. Thus

$$E(G,s) = -\sum_{a \in A'} 4a = -4B \leq K.$$

Now consider a sequence $s$ with $\lceil \lambda n \rceil$ or fewer hydrophobics for which $E(G,s) \leq K$. Let $n_H$ equal the number of hydrophobics in $s$. The maximum number of contacts each hydrophobic can make is two, so $E(G,s) \geq -n_H \geq -\lceil n\lambda \rceil = -4B = K$. Now $E(G,s) \leq K$, so $n_H = \lceil n\lambda \rceil$. The contact graph contains connected components that are even length cycles with lengths $4a$, $a \in A$. To guarantee that each hydrophobic contributes exactly one to the total energy, the hydrophobics have to be used to fill cycles in the contact graph; if any cycle is filled only partially, there will exist a hydrophobic that only has one contact, from which it follows that $E(G,s) > K$, which is a contradiction. Now let $A'$ represent the set of elements in $A$ that correspond to cycles that are completely filled. It follows that

$$\sum_{a \in A'} a = -E(G,s)/4 = B.$$

We have shown that an optimal sequence for $G$ has energy $K$ if and only if there is a subset of $A$ with size $B$. Now the transformation from SUBSET SUM requires polynomial time, and we can quickly verify whether or not a given sequence has energy less than or equal to $K$. Thus we conclude that $(L, \lambda, HP)$-IPF is NP-complete. This argument utilizes embeddings into the 2D cubic lattice. Since this is a subset of the 3D lattice, this argument applies for both the 2D and 3D cubic lattices. ■