The Inapproximability of Side-Chain Positioning

Bernard Chazelle[*]
chazelle@cs.princeton.edu

Carl Kingsford[†]
carlk@cs.princeton.edu

Mona Singh[‡]
mona@cs.princeton.edu

Abstract

Side-chain positioning is a central component of homology modeling and protein design. In a common formulation of the problem, the backbone is fixed, side-chain conformations come from a rotamer library, and a pairwise energy function is optimized. In [5], we showed that it is NP-complete to find even a reasonable approximate solution to this problem. Here, we explain, for non-computer scientists, the result in more detail.

1 Introduction

Side-chain positioning (SCP) is a key step in computational methods for predicting and designing protein structures. A widely-studied formulation of the problem assumes a rigid backbone, a pairwise energy function, and a set of possible rotamer choices for each C_α position on the backbone. The goal is to choose a rotamer for each position to find the global minimum energy conformation (GMEC). This formulation of SCP has been the basis of some of the more successful methods for homology modeling (e.g., [21, 24, 13, 4]), and protein design (e.g., [6, 20, 19]).

It has been shown that SCP is NP-complete [23]. For any NP-complete problem, if there is an efficient (i.e., polynomial-time) algorithm for it, then that algorithm can be used to solve all problems in the complexity class NP efficiently. The class NP includes all the problems whose solutions can be certified efficiently. For example, the problem of finding if a set of N numbers has a subset that sums to 1 is in NP because any candidate solution, i.e., a subset of the given numbers, can be certified quickly: simply add them up. Predictably, most of the problems encountered in biology are in NP. Many of the interesting ones are widely believed not to have efficient algorithms for finding optimal solutions. Accordingly, since SCP is NP-complete, it is unlikely that there is any efficient algorithm for solving the problem optimally.

For many NP-complete problems (including, for example, Max-Cut[1] [8]), it is possible to develop approximation algorithms that can efficiently find a suboptimal solution that is within a provable factor of the optimal one. Approximation algorithms are different from heuristics, which are algorithms that should work well in many instances, but have no performance guarantee on the quality of the solution. In other words, for NP-complete problems that permit approximations, efficient algorithms exist for finding provably "good," though not optimal, solutions. For SCP, we have shown that it is unlikely that there is any approximation algorithm with a reasonable performance guarantee. In particular, in [5], we showed:

Theorem 1.1 *It is NP-complete to approximate the minimum energy of the GMEC within a factor of cn, where c is a positive constant and n is the total number of rotamers.*

[*] Department of Computer Science, Princeton University. Supported in part by NSF grant CCR-998817, ARO Grant DAAH04-96-1-0181, and NEC Research Institute.

[†] Department of Computer Science and Lewis-Sigler Institute for Integrative Genomics, Princeton University

[‡] Department of Computer Science and Lewis-Sigler Institute for Integrative Genomics, Princeton University. Supported in part by NSF PECASE Grant MCB-0093399 and DARPA grant MDA972-00-1-0031.

[1] Max-Cut is the problem of partitioning the nodes of a graph into two non-empty sets so that the number of edges between the two sets is as large possible.

Some terms in the above statement need clarification. An algorithm is said to "approximate within a factor cn" if, for any input, the algorithm produces a choice of rotamers such that the energy of that choice is no more than cn times the energy of the optimal choice of rotamers. Our theorem indicates that such an algorithm is unlikely to be efficient. Another detail is that complexity results are proved for yes/no decision questions. The SCP problem is an optimization problem in which we are given an instance of a side-chain positioning problem, and we seek the best conformation as well as its energy. It is turned into a yes/no decision problem by providing as additional input an integer k, and asking whether the GMEC of the instance has energy less than k. Note that this modified problem is not harder than the original optimization version: if one could solve the optimization version, one could easily solve this yes/no decision version.

Theorem 1.1 does not mean that good algorithms and methods for the side-chain positioning problem cannot be shown to work well in practice. Indeed, several papers have presented efficient algorithms that seem to work well in practice (e.g. [24, 4]), or algorithms that are designed to find optimal solutions but complete quickly for some problems (e.g. [9, 15, 7, 22, 14]).

Section 2 gives a more formal definition of the side-chain positioning problem. Section 3 gives the proof of Theorem 1.1.

2 Formulation

The SCP problem can be stated as follows [7]. Given a fixed backbone of length p, each residue position i is associated with a set of possible candidate rotamers $\{i_r\}$. Once a single rotamer for each residue position has been chosen, the energy of a protein system is given by the formula

$$\mathcal{E} = E_0 + \sum_i E(i_r) + \sum_{i<j} E(i_r j_s),$$

where E_0 is the self-energy of the backbone, $E(i_r)$ is the energy of the interaction between the backbone and the chosen rotamer i_r at position i as well as the intrinsic self-energy of rotamer i_r, and $E(i_r j_s)$ is the pairwise interaction energy between chosen rotamers i_r and j_s. We seek an assignment of rotamers to positions that minimizes the overall energy of the system.

It is convenient to reformulate the SCP problem in graph-theoretic terms. Let G be an undirected p-partite graph with node set $V_1 \cup \cdots \cup V_p$, where V_i includes a node u for each rotamer i_r at position i; the V_i's may have varying sizes. Each node u of V_i is assigned a weight $E_{uu} = E(i_r)$; each pair of nodes $u \in V_i$ and $v \in V_j$ $(i \neq j)$, corresponding to rotamers i_r and j_s respectively, is joined by an edge with a weight of $E_{uv} = E(i_r j_s)$. Zero-weight edges can be thought of as equivalent to the absence of an edge, and the node weights can be modeled as self-loop edges. The GMEC is achieved by picking one node per V_i to minimize the weight of the induced subgraph.

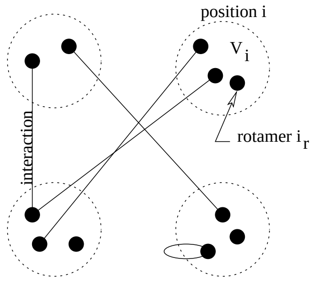

Figure 1: The graph formulation. In this hypothetical example, there are four positions in the protein, one with two rotamer possibilities and the others with three rotamer possibilities each.

3 Proof of Theorem 1.1

We will prove this theorem by showing that if the SCP problem has a good approximation algorithm then we would also have an efficient algorithm for a problem for which it is likely that none exists, namely a problem involving satisfiability of boolean formulas.

A 3-CNF formula is a conjunction of clauses, each one consisting of the disjunction of three literals (not necessarily distinct). An example of such a boolean formula is shown at the top of Figure 2. In that figure, letters $a, b, \ldots$ represent variables and $\bar{a}, \bar{b}, \ldots$ represent the negation of those variables. $\vee, \wedge$ represent "or" and "and," respectively. A formula is *satisfiable* if the variables can be assigned values **true** or **false** such that the whole formula is true. Given a 3-CNF formula, it is NP-complete to determine whether it is satisfiable.

The PCP theorem [2, 3] asserts that, given any 3-CNF formula Φ on n variables, there exists another one, denoted by Ψ, which contains $n^{O(1)}$ variables and is satisfiable if and only if Φ is satisfiable. Furthermore, if

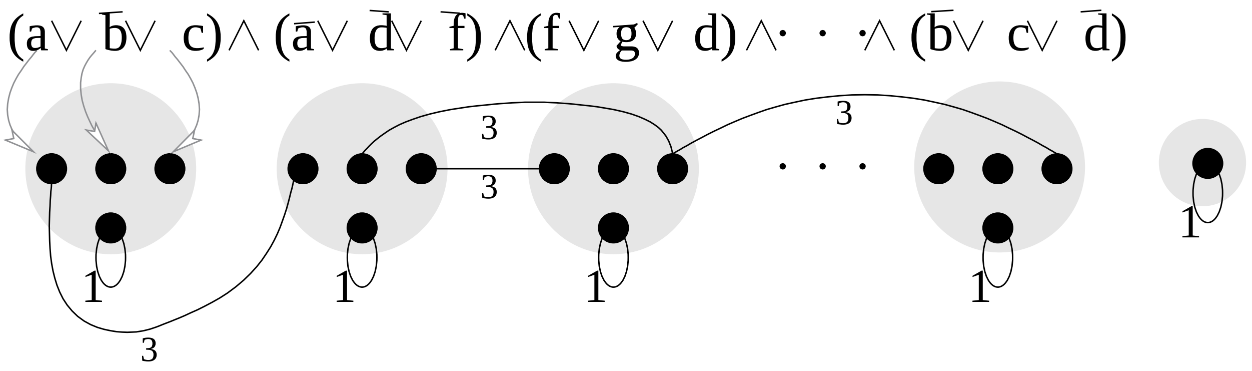

Figure 2: Converting a 3CNF formula to a SCP problem.

Ψ is not satisfiable, then it is *strongly unsatisfiable*, meaning that no truth assignment can satisfy more than a fraction α of its clauses, for some constant $0 < \alpha < 1$. Finally, Ψ can be derived from Φ in polynomial time. Since 3-CNF satisfiability is NP-complete, it is then also NP-complete to distinguish between formulas that are satisfiable and those that are strongly unsatisfiable. (If there is an efficient algorithm that distinguishes between such formulas, then any 3-CNF formula can be efficiently tested for satisfiability by first converting it to a strongly unsatisfiable formula and then using this algorithm.)

Given a 3-CNF formula with p clauses that is either satisfiable or strongly unsatisfiable, we create an SCP problem such that if the formula is satisfiable then the GMEC = 1, but if the formula is not satisfiable then the GMEC will tell us how many clauses can be satisfied in the original 3-CNF.

We build a $p+1$-partite graph G as follows: each clause i corresponds to a set V_i of 4 vertices. In each V_i three vertices are associated with the literals of clause i. These vertices have no self-weights. Two vertices in V_i and V_j are joined in G if and only if the literal of one is the negation of the other. Each such edge is assigned weight 3. The 4th vertex in each V_i is an "extra" vertex with no adjacent edges and vertex weight 1. We add an additional position with a single node of weight 1. The total number of rotamers, or nodes, in this SCP instance is $n = 4p + 1$. This reduction is depicted in Figure 2.

If the CNF formula is satisfiable then for each V_i we select a literal set to true as the GMEC vertex. These p vertices form an independent set (i.e., since one cannot set both a variable and its negation to true, these vertices have no edges between them) and the energy of the system is 1.

If the CNF formula is not satisfiable then the GMEC is formed by picking the largest independent set among the vertices, including at most one vertex per V_i, and completing the selection for the remaining V_i by choosing the fourth "extra" vertex for each. (Picking any pair of adjacent vertices would be a mistake since that choice could be locally improved by choosing an isolated vertex in each position of weight 1.) We can set to true the literals corresponding to the vertices of the independent set. Therefore, the energy of the GMEC is $p - c + 1$, where c is the maximum number of satisfiable clauses in the CNF formula. Because the CNF formula is strongly unsatisfiable, the minimum number of unsatisfied clauses $p - c$ is at least $(1 - \alpha)p$, and the optimal GMEC of the corresponding SCP problem is at least $(1 - \alpha)p + 1$.

Thus, suppose we had an efficient algorithm that was guaranteed to find a solution $< \frac{(1-\alpha)}{4}n$ times the optimal. Then for any satisfiable formula, this algorithm would find a solution to the corresponding SCP problem of value at most $\frac{(1-\alpha)}{4}n$. Since this is less than $(1 - \alpha)p + 1$, the minimum possible value of the GMEC corresponding to an unsatisfiable formula, this algorithm could distinguish between satisfiable and strongly unsatisfiable formulas, something the PCP Theorem implies we cannot do.

4 Discussion

While it is NP-complete to find even an approximate solution to the side-chain positioning problem, in practice large instances of SCP have been solved using both exhaustive and heuristic techniques (e.g., [7, 9, 15, 11, 1, 18, 10, 4, 17, 12, 16, 24]). This NP-completeness result describes worst-case behavior, and it may not hold for the classes of problems and energy functions that occur in practice.[2] Indeed, in [14], we used linear programming to probe instances of side-chain positioning, and have shown empirically that in many interesting cases, it is often possible to find optimal solutions in polynomial-time. Our analysis suggests that solutions to homology modeling problems are easier to find than those for protein design. An intriguing open question is to uncover what features of the side-chain positioning problem can make the problem easy or hard to solve in practice, and whether these features suggest an alternative formulation of the problem.

References

[1] E. Althaus, O. Kohlbacher, H.-P. Lenhof, and P. Müller. A combinatorial approach to protein docking with flexible side-chains. In *Proceedings 4th Annual International Conference on Computational Molecular Biology*, pages 15–24, 2000.

[2] S. Arora, C. Lund, R. Motwani, M Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[3] S. Arora and M. Safra. Probabilistic checking of proofs: A new characterization of np. *J. ACM*, 45(1):70–122, 1998.

[4] Michael J. Bower, Fred E. Cohen, and Roland L. Dunbrack, Jr. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: A homology modeling tool. *J. Mol. Biol.*, 267:1268–1282, 1997.

[5] Bernard Chazelle, Carl Kingsford, and Mona Singh. A semidefinite programming approach to side-chain positioning with new round ing strategies. *INFORMS J. Computing*, 16:380–392, 2004.

[6] Bassil I. Dahiyat and Stephen L. Mayo. De novo protein design: Fully automated sequence selection. *Science*, 278:82–87, October 1997.

[7] Johan Desmet, Marc De Maeyer, Bart Hazes, and Ignace Lasters. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356:539–542, April 1992.

[8] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995.

[9] Robert F. Goldstein. Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophys. J.*, 66:1335–1340, 1994.

[10] D. B. Gordon, G. Hom, S. Mayo, and N. Pierce. Exact rotamer optimization for protein design. *J. Comput. Chemistry*, 24:232–243, 2002.

[11] D. B. Gordon and Stephen L. Mayo. Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *J. Comput. Chem.*, 19(13):1505–1514, 1998.

[12] S. Holm and C. Sander. Database algorithm for generating protein backbone and sidechain coordinates from a Ca trace: Application to model building and detection of coordinate errors. *J. Mol. Biol.*, 218:183–194, 1991.

[13] T. Alwyn Jones and Gerard J. Kleywegt. CASP3 comparative modeling evaluation. *Proteins*, 37:30–46, 1999.

[14] Carl Kingsford, Bernard Chazelle, and Mona Singh. Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics*, 2004. Advance access publication on 11/16/2004.

[2] As an example, if the energy function obeys the triangle inequality, it is easy to show that it is possible to obtain a 2-approximation. However, such an energy function is not realistic for either homology modeling or protein design problems.

[15] I. Lasters, M. De Maeyer, and J. Desmet. Enhanced dead-end elimination in the search for the global minimum energy conformation of a collection of protein side chains. *Prot. Eng.*, 8:815–822, 1995.

[16] Andrew R. Leach and Andrew P. Lemon. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. *Proteins*, 33:227–239, 1998.

[17] C. Lee and S. Subbiah. Prediction of protein side-chain conformation by packing optimization. *J. Mol. Biol.*, 217(2):373–388, January 1991.

[18] L. L. Looger and H. W. Hellinga. Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: implications for protein design and structural genomics. *J. Mol. Biol.*, 307(1):429–445, 2001.

[19] Loren L. Looger, Mary A. Dwyer, James J. Smith, and Homme W. Hellinga. Computational design of receptor and sensor proteins with novel functions. *Nature*, 423:185–190, May 2003.

[20] S. M. Malakauskas and S. L. Mayo. Design, structure and stability of a hyperthermophilic protein variant. *Nat. Struct. Biol.*, 5(6):470–475, June 1998.

[21] D. Petrey, Z. Xiang, C. Tang, L. Xie, M. Gimpelev, T. Mitros, C. Soto, S. Goldsmith-Fischman, A. Kernytsky, A. Schlessinger, I. Koh, E. Alexov, and B. Honig. Using multiple structure alignments, fast model building and energetic analysis in fold recognition and homology modeling. *Proteins*, 53:430–435, 2003.

[22] N. A. Pierce, J. A. Spriet, J. Desmet, and S. L. Mayo. Conformational splitting: A more powerful criterion for dead-end elimination. *J. Comput. Chem*, 21(11):999–1009, 2000.

[23] N. A. Pierce and E. Winfree. Protein design is NP-hard. *Prot. Eng.*, 15(10):779–782, October 2002.

[24] Zhexin Xiang and Barry Honig. Extending the accuracy limits of prediction for side-chain conformations. *J. Mol. Biol.*, pages 421–430, 2001.