A Semidefinite Programming Approach to Side-Chain Positioning with New Rounding Strategies

Bernard Chazelle*
Department of Computer Science
Princeton University and
NEC Research Institute
chazelle@cs.princeton.edu

Carl Kingsford
Department of Computer Science and
Lewis-Sigler Institute for Integrative Genomics
Princeton University
carlk@cs.princeton.edu

Mona Singh†
Department of Computer Science and
Lewis-Sigler Institute for Integrative Genomics
Princeton University
mona@cs.princeton.edu

Abstract

Side-chain positioning is an important subproblem of the general protein structure prediction problem, with applications in homology modeling and protein design. The side-chain positioning problem takes a fixed backbone and a protein sequence and predicts the lowest energy conformation of the protein's side-chains on this backbone. We study a widely-used version of the problem where the side-chain positioning procedure uses a rotamer library and an energy function that can be expressed as a sum of pairwise terms. The problem is NP-complete; we show here that it cannot even be approximated. In practice, it is tackled by a variety of general search techniques and specialized heuristics. Here, we propose formulating the side-chain positioning problem as an instance of semidefinite programming (SDP). We introduce two novel rounding schemes and provide theoretical justifications for their effectiveness under various conditions. We apply our method on simulated data, as well as on the computational

*Supported in part by NSF grant CCR-998817, ARO Grant DAAH04-96-1-0181, and NEC Research Institute.

†Supported in part by NSF PECASE Grant MCB-0093399 and DARPA grant MDA972-00-1-0031.

redesign of two naturally occurring protein cores, and show that our SDP approach generally finds good solutions. Beyond the context of side-chain positioning, our very general rounding schemes should be applicable elsewhere.
(*Computational Biology; Semidefinite Programming; Side-Chain Positioning*)

1. Introduction

A central problem in molecular biology is that of predicting a protein's three-dimensional fold when given only its one-dimensional amino acid sequence. This is an important problem since the structure of a protein plays a critical role in its function, and while the number of known protein sequences is growing rapidly, their corresponding protein structures are being determined at a significantly slower pace. Despite decades of work, the problem of predicting the 3D structure of a protein from its amino acid sequence remains unsolved. Here, we consider the *side-chain positioning (SCP)* problem, a challenging and important component of the general protein structure prediction problem.

Further Background. A protein molecule is formed from a chain of amino acids. Each amino acid consists of a central carbon atom, and attached to this carbon are a hydrogen atom, an amino group (NH_2), a carboxyl group (COOH) and a *side-chain* that characterizes the amino acid. Side-chains vary in composition; for example, the side-chain for the amino acid Glycine consists of a single hydrogen atom, and the side-chain for the amino acid Alanine consists of a carbon atom with three hydrogen atoms attached. The amino acids of a protein are connected in sequence with the carboxyl group of one amino acid forming a peptide bond with the amino group of the next amino acid. This forms the *protein backbone*, and the repeating amino acid units within the protein (also called *residues*) consist of both the main-chain atoms that comprise the backbone as well as the side-chain atoms.

There are 20 commonly occuring amino acids, and each protein molecule is specified by a sequence corresponding to the amino acids that make it up. Whereas a protein's sequence immediately reveals its chemical composition, its structure is significantly more difficult to determine. The structure of a protein is specified by the coordinates of its main-chain and side-chain atoms, and it is generally believed that a protein's native structure corresponds to its global free energy minimum. A common approach to predict protein structure computationally is thus to start with the protein's amino acid sequence, specify an appropriate energy function, and find the conformation that minimizes the energy function. Protein structures are difficult to predict due to inaccuracies in energy functions as well as the infeasibility of computationally searching over all possible conformations; in practice, predictions are often made by settling for less than optimal solutions when considering imperfect energy functions.

In this paper, we focus on the computational issues involved in protein structure prediction and consider the problem where the structure of a protein's backbone is known, and the

goal is to predict the coordinates of its its side-chains atoms. More specifically, in SCP, one is given a fixed backbone and a protein sequence, and the task is to predict the best conformation of the protein's side-chains on the backbone. The problem is made discrete by the observation that in actual protein structures, side-chains tend to occupy one of a small number of conformations [42], called *rotamers*. These rotamers are identified by finding frequently occurring side-chain conformations in databases of protein structures, and common conformations for each amino acid side-chain are collected into rotamer libraries [16]. Furthermore, the total energy of the molecule is expressed as a sum of pairwise energies between atoms (i.e., when computing energies, only two atoms are considered at a time). SCP can then be formulated as a combinatorial optimization problem: choose a rotamer for each side-chain such that the overall energy of the molecule is minimized (see Section 2).

Applications. This formulation of SCP is the basis of some of the more successful methods for *protein design* [12] and *homology modeling* [34]. In protein design, the goal is to find the sequence of amino acids that will fold into a given shape. This is often reduced to SCP by the following method: rather than specifying exactly the amino acid at each position, we allow the optimization problem to choose among rotamers from *several* different types of amino acids at each position. The optimization problem is solved and the amino acid that corresponds to the rotamer that was chosen at position i is taken to be the ith amino acid in the sequence. This sequence is the one that best fits this backbone, and thus, it is hoped, will fold into this shape. This approach has led to some dramatic successes in protein design, including the design of a 28 residue zinc finger domain that folds in the absence of zinc [12]. We computationally redesign two naturally-occurring proteins in Section 4.

Homology modeling is used to predict the structure of a protein when there is another protein of known structure with which it shares high sequence similarity. In this case, the two proteins almost always have a similar overall shape, and thus the protein of known structure can provide a reasonable template backbone for the protein under investigation. The fixed-backbone formulation of SCP we study here is the basis of several widely-used and successful homology modeling packages (e.g., [10]).

Related Work. SCP is NP-complete [41], and, as we show here, inapproximable. However, there has been progress on both exhaustive and heuristic techniques for this problem. Within the past dozen years, a series of papers on "dead-end elimination" (DEE) [14, 13, 23, 29, 24] have given simple rules for throwing out rotamers that cannot possibly be in the optimal solution. Special purpose heuristic search techniques for specific energy functions have been successfully applied (e.g., as in the SCRWL package [10]). More general search methods such as simulated annealing (e.g., [34]), A* [32] and mean-field-optimization [33] have also been applied.

More recently, the side-chain positioning problem has been formulated as an integer linear program [3, 17], where a relaxed linear program (LP) is used as a subroutine to find optimal solutions to the problem, using either branch-and-bound or branch-and-cut.

Main contributions. We characterize the complexity of SCP by showing that it is NP-hard to approximate the minimum energy to within a factor of cn, where n is the total number of possible rotamers and c is a positive constant; that is, in the worst case, it is hard to find even an approximate solution with any kind of theoretical guarantee. This result is given in Section 5.

The crux of this paper describes a semidefinite programming (SDP) heuristic for SCP. In contrast to the provable hardness of the problem, empirical studies show that, in practice, our SDP approach finds good solutions. By relying on SDP, we can take advantage of the extensive research directed at the general problem, including off-the-shelf solvers. The methods that use LP have a similar advantage; however, those methods are typically used in the context of branch-and-bound and branch-and-cut, whereas our approach runs in polynomial time.

Our method works in three steps: first, relax the SCP problem into an instance of SDP; next, solve it in polynomial time by an interior-point method; finally, convert the solution into 0/1 form by randomized rounding [43, 44]. This general approach for approximation algorithms was pioneered by Lovász's ground-breaking work on the ϑ function [25, 36] and Goemans and Williamson's ingenious MAX-CUT [22] algorithm, and it has been pursued further since (e.g., [20, 2, 18, 27, 8, 47]).

In order to convert the fractional SDP solutions into rotamer choices for the original SCP problem, we introduce two new techniques for randomized rounding. These are general techniques that may have applicability beyond the SCP problem. The first technique, *projection rounding*, is based on the geometry of the solution vectors, and the second, *Perron-Frobenius rounding*, is based on spectral properties of the solution matrix. Our Perron-Frobenius rounding scheme approximates the solution matrix by the eigenvector corresponding to its highest eigenvalue. This is, of course, a standard trick (see, e.g., [15, 9, 7]); however, our Perron-Frobenius rounding is different in that we decompose a matrix that does not have a graph-theoretic interpretation, and we rely crucially on the positivity of the entries of its highest eigenvector.

In light of the inapproximability result, no rounding scheme should have good performance on all instances; however, we provide some theoretical justification for good performance on some types of input. We argue that under various assumptions about the statistical nature of the problem, the expected difference (the drift) between the total energies given by the optimal fractional solution and our randomized rounding integral solutions is small.

We have applied our method to redesign computationally the cores of two naturally occurring proteins, the *Bacillus caldolyticus* cold shock protein and the TIM barrel triose

phosphate isomerase from chicken. We have also experimented successfully on general random graphs as well as a class of random graphs that better capture the geometry of actual proteins. Since LP-based approaches for SCP are effective in practice [3], we compare our method to LP; this comparison highlights the benefits of SDP's additional computational machinery. Our empirical studies show that, in practice, good solutions to SCP are found by our two randomized rounding schemes. Additionally, we note that since SDP provides better lower bounds than LP for the underlying SCP problem, it is a more effective bounding function for branch-and-bound or branch-and-cut [17, 3] approaches.

Independently, Lau [30] and Lau and Watanabe [31] applied semidefinite programming and randomized rounding to the more restricted problem of weighted constraint satisfaction; this is a special case of the SCP problem considered here. They also give an inapproximability result that is weaker than ours in the general case.

At present, SDP solvers are limited to solving small problem sizes. However, as SDP approaches are increasingly being applied to combinatorial optimization problems, SDP solvers continue to improve. As larger proteins and rotamer libraries are considered, exhaustive techniques (such as branch-and-bound or A*) may be limited by their potentionally exponential running time. In constrast, our semidefinite programming approach runs in polynomial time, and the approaches developed in this paper, which we show work well on problems of interest, will have broad applicability. Finally, as opposed to other heuristic techniques (such as simulated annealing), as more is discovered about the nature of SCP applications in practice, our SDP formulation permits the development of other rounding schemes that better exploit the real-world statistical properties of the problem.

2. Problem Formulation

The version of the side-chain positioning problem we study is as follows: given a backbone, a protein sequence, a rotamer library, and a pairwise energy function, choose a rotamer for each amino acid side-chain such that the overall energy is minimized. More formally, the SCP problem can be stated as follows [14]. Given a fixed backbone of length p, each residue position i is associated with a set of possible candidate rotamers $\{i_r\}$. In the design problem, this set may include rotamers from several kinds of amino acids. 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 resulting from the interaction between the backbone and the chosen rotamer i_r at position i as well as the intrinsic energy of rotamer i_r, and $E(i_r j_s)$ accounts for the pairwise interaction energy between chosen rotamers i_r and j_s. In this discretized setting, the placement of each side-chain is reduced to finding an assignment of rotamers to positions that minimizes the overall energy of the system. This assignment is called the

global minimum energy conformation or *GMEC*.

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.

3. A Semidefinite Programming Heuristic

We present in this section a formulation of the SCP problem as a semidefinite program. Given a graph G with node set $V = V_1, \ldots, V_p$, assign to each $u \in V$ a 0/1 variable x_u. The intuition is that x_u will be 1 if rotamer u is selected, 0 otherwise. Computing the GMEC is equivalent to solving the following integer quadratic programming problem:

$$\text{Minimize} \qquad \sum_{(u,v)\in G} E_{uv} x_u x_v \tag{1}$$

$$\text{subject to} \qquad \sum_{u\in V_i} x_u = 1 \quad \text{for } i = 1, \ldots, p \tag{1a}$$

$$x_u \in \{0, 1\}\,.$$

We rewrite (1) into a form that will be more convenient to relax into a semidefinite program with as few constraints as possible. Add a new position with an isolated vertex u_0 to G and define its singleton vertex set $V_0 = \{u_0\}$. The constraints (1a) applied to this position imply $x_{u_0} = 1$. We square both sides of (1a) and, using the fact that $x_u \in \{0, 1\}$, we add two new sets of constraints to obtain the equivalent program

$$\text{Minimize} \qquad \sum_{(u,v)\in G} E_{uv} x_u x_v \tag{2}$$

$$\text{subject to} \qquad \sum_{u,v\in V_i} x_u x_v = 1 \quad \text{for } i = 0, \ldots, p$$

$$\sum_{u\in V_i} x_{u_0} x_u = 1 \quad \text{for } i = 0, \ldots, p$$

$$x_u x_u = x_{u_0} x_u \quad \text{and} \quad x_u \in \{0, 1\} \quad \text{for all } u\,.$$

The relaxation step lifts each x_u to $\mathbb{R}^n$, where n is the number of nodes in G (including the dummy node), scalar multiplication is replaced by the dot product, and the requirement $x_u \in \{0, 1\}$ is replaced by $0 \leq x_u^T x_v \leq 1$, for all u and v. Quadratic programming is NP-hard

in general, but this relaxed system is an instance of positive semidefinite programming, and it can be solved efficiently. To see that, we linearize all the constraints by introducing the variable x_{uv} to denote $x_u^T x_v$. To ensure that this linearization is *not* a relaxation, we require that the n-by-n matrix $X = (x_{uv})$ be positive semidefinite (PSD). We also note that the constraints $x_u^T x_v \leq 1$ are redundant since X is PSD and the diagonal elements are ≤ 1. Thus, we get:

$$\begin{aligned} \text{Minimize} \quad & \sum_{(u,v)\in G} E_{uv}\, x_{uv} \\ \text{subject to} \quad & x_{uu} = x_{u_0 u} \quad \text{and} \quad x_{uv} \geq 0 \\ & \sum_{u\in V_i} x_{u_0 u} = \sum_{u,v\in V_i} x_{uv} = 1 \quad \text{for } i = 0, \ldots, p \\ & X \text{ is PSD}. \end{aligned} \tag{3}$$

We can solve the SDP system (3) in polynomial time to within any level of accuracy by using the ellipsoid algorithm (see e.g. [25]) or, preferably, an interior-point method (see e.g. [1, 39, 46]).

Next, we must map each vector x_u to $\widehat{x}_u \in \{0, 1\}$ so that $\sum_{u\in V_i} \widehat{x}_u = 1$ and so that ideally the expected increase in the value of the objective function, the *drift*, is small. We discuss two rounding schemes, both of which fit the basic format of *randomized rounding* [43]. The idea is to specify a probability distribution for each position and home in on a solution by sampling from it. We describe two distributions, one based on projection, the other on spectral approximation. The first one is very simple and easy to implement; the second one requires only slightly more work. See Section 4 for some empirical comparisons of the two rounding methods. The following characterization of the geometry of the solution vectors will be useful when we discuss the rounding schemes.

Lemma 3.1 *If $X = (x_{uv})$ is a solution to (3) where $x_{uv} = x_u^T x_v$ for vectors x_u, x_v, then all the vectors $\sum_{u\in V_i} x_u$ are equal to x_{u_0}, and each x_u belongs to the unit-diameter sphere with antipodes O (the origin) and x_{u_0}.*

Proof: Fix $i \geq 0$ and let $y = \sum_{u\in V_i} x_u$. The constraints imply that $\|y\|_2^2 = \sum_{u,v\in V_i} x_{uv} = 1$. Meanwhile, the inner product of y and x_{u_0} is equal to $\sum_{u\in V_i} x_{u_0 u} = 1$. We also have $x_{u_0}^T x_{u_0} = 1$. Since their lengths are the same and equal to the projections onto one another, it follows that $y = x_{u_0}$ for all i. Now, take any node $u \in V_i$. Observe that

$$\left\| x_u - \frac{x_{u_0}}{2} \right\|_2^2 = x_{uu} - x_{u_0 u} + \frac{1}{4} = \frac{1}{4},$$

where $x_{uu} = x_{u_0 u}$ follows directly from the constraints. Therefore, x_u belongs to the sphere centered at $x_{u_0}/2$ of radius $1/2$. This sphere passes through the two points O and x_{u_0}, which are antipodal. □

We will compare our semidefinite program to the LP relaxation of the following IP:

$$\begin{aligned} \text{Minimize} \quad & \sum_{(u,v)\in G} E_{uv}\, x_{uv} \\ \text{subject to} \quad & \sum_{u\in V_i} x_{uu} = 1 \quad \text{for } i = 1,\ldots,p \\ & \sum_{u\in V_i} x_{uv} = x_{vv} \quad \text{for } i = 1\ldots,p \text{ and any } v \\ & x_{uv} \in \{0,1\} \end{aligned} \tag{4}$$

This LP formulation of SCP is similar to those proposed by Eriksson et al. [17] and Althaus et al. [3]. The benefit of our SDP formulation over the LP formulation is two-fold: first, our relaxation is more constrained so its solution is closer to that of the integer program. The SDP formulation generates second moments between the nodes [8], and our Perron-Frobenius rounding scheme will implictly make use of them. Second, the solutions are vectors and not scalars. This gives us much more freedom in the rounding phase of the algorithm and allows for effective use of the "geometry" of the problem. We will compare our semidefinite program to this IP and LP in Section 4.

3.1 Projection Rounding

This scheme is based on the fact that the constraints guarantee that $\sum_{u\in V_i} \|x_u\|_2^2 = 1$ for any $1 \le i \le p$, so that the quantities $q_u = \|x_u\|_2^2$ associated with the nodes u of V_i form a valid probability distribution from which we can sample effectively.

- THE ROUNDING RULE: For each $1 \le i \le p$, choose $u \in V_i$ at random with probability q_u.

Note that only one u is chosen per V_i. This is called projection rounding because the probability of choosing u is equal to $x_{uu} = x_{u_0u}$, which is the length of the projection of x_u onto x_{u_0}. By looking at the geometry of the SDP formulation in a manner similar to [2, 18, 27], we can provide a measure of theoretical justification for our rounding strategy.

We first provide some intuition behind this rounding scheme. The solution vectors are constrained to lie on a sphere and the projection rounding rule favors choosing long vectors. If a single vector x_u is dominant within its V_i — a common occurrence — then simple geometry (Figure 1a) shows the dot products of these vectors should also be big. Because the solution matrix is positive semidefinite, an off-diagonal element x_{uv} is the dot product of the two vectors x_u and x_v, and for long vectors x_u, x_v we can expect x_{uv} to be large as well. We can thus hope to avoid the most damaging situation: where the rounding scheme chooses nodes u and v but the fractional solution has put low or zero weight on the edge between them. The intuition holds in the opposite case as well: two low probability vectors

(that is, short vectors), are likely to have a small dot product, and we would like to avoid choosing the edge that corresponds to that dot product. We develop this argument more formally below when we give an upper bound on the drift, defined as the expected difference between the post- and pre-rounding objective function value.

Let $\widehat{x}_u$ be 1 if u is chosen in the rounding stage and 0 otherwise. As usual, $\{x_u\}$ denotes the solution of the (relaxed) SDP system. The expected value of the objective function, post-rounding, is equal to

$$\mathbf{E}\Big\{ \sum_{(u,v)\in G} E_{uv}\, \widehat{x}_u \widehat{x}_v \Big\} = \sum_u E_{uu} \|x_u\|_2^2 \; + \sum_{(u,v)\in G_o} E_{uv} \|x_u\|_2^2 \|x_v\|_2^2,$$

where G_o denotes the set of nonloop edges in G. Thus, the drift Δ is

$$\Delta = \sum_{(u,v)\in G_o} E_{uv} \Big(\|x_u\|_2^2 \|x_v\|_2^2 - x_u^T x_v \Big) .$$

Observe that the drift originates exclusively from the off-diagonal entries. Let y_u denote the projection of x_u on the orthogonal complement $x_{u_0}^{\perp}$ of x_{u_0}. We rewrite the drift in terms of these y_u. Because x_{u_0} is of unit length, we have

$$x_u^T x_v = ((x_u^T x_{u_0}) x_{u_0} + y_u)^T ((x_v^T x_{u_0}) x_{u_0} + y_v) = \\ (x_u^T x_{u_0})(x_v^T x_{u_0}) + y_u^T y_v = \|x_u\|_2^2 \, \|x_v\|_2^2 + y_u^T y_v \, ;$$

therefore,

$$\Delta = - \sum_{(u,v)\in G_o} E_{uv}\, y_u^T y_v \, . \tag{5}$$

In the special case where all the energies are non-negative, it is also possible to relate the drift directly to the lengths of the x_u's. (While this is not true for all energy functions, the popular side-chain positioning package SCWRL [10] only has non-negative energies.) By Lemma 3.1 and the Pythagorean theorem applied to the right triangle in Figure 1b,

$$\|y_u\|_2^2 + \Big(\|x_u\|_2^2 - \frac{1}{2} \Big)^2 = \|y_u\|_2^2 + \Big(x_u^T x_{u_0} - \frac{1}{2} \Big)^2 = \frac{1}{4} \, .$$

It follows that $\|y_u\|_2 = \|x_u\|_2 \sqrt{1 - \|x_u\|_2^2}$. Assuming nonnegative energies, by Cauchy-Schwarz,

$$\Delta \le \sum_{(u,v)\in G_o} E_{uv} \, |y_u^T y_v| \le \sum_{(u,v)\in G_o} E_{uv} \, \|x_u\|_2 \|x_v\|_2 \sqrt{(1 - \|x_u\|_2^2)(1 - \|x_v\|_2^2)}. \tag{6}$$

The Sharp-Concentration Case Our algorithm is expected to do very well when, within each V_i, the probability distribution is sharply concentrated. In other words, if the probability $\|x_u\|_2^2$ of picking u greatly exceeds that of selecting the other vertices $v \in V_i$, then projection rounding does the right thing. Indeed, if within each V_i one $\|x_u\|_2$ is near 1, then the other $\|x_v\|_2$'s ($v \in V_i$) must be small. This implies that the product $\|x_u\|_2 \sqrt{(1 - \|x_u\|_2^2)}$ is always small and, by (6), so is the drift.

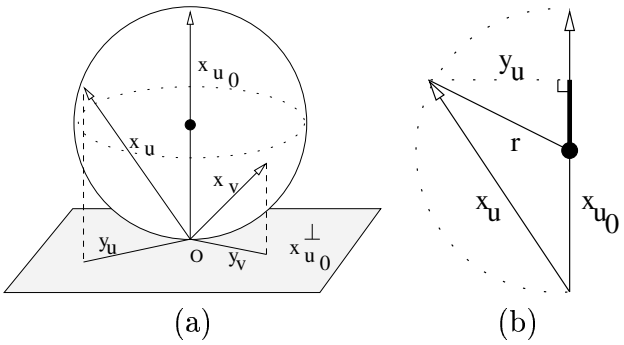


Figure 1: The solution vectors are constrained to lie on the sphere. Equation (5) bounds the drift of projection rounding in terms of the projections of the solution vectors onto the hyperplane orthogonal to x_{u_0}.

3.2 Perron-Frobenius Rounding

Algebraically, projection rounding entails approximating $X = W^T W$ by the rank-one matrix $\widehat{X} = W^T x_{u_0} x_{u_0}^T W$. Are there better low-rank approximation matrices? To answer this question, we return to the SDP formulation (3), which ensures that the matrix X is nonnegative. Because X is also positive semidefinite a spectral approach suggests an alternative way: approximate X by a rank-one matrix so that the difference has minimum L^2 norm.

To simplify the notation, we move all the energies over to the edges by defining $F_{uv} = E_{uv} + \frac{1}{p-1}(E_{uu} + E_{vv})$ if $u < v$, and 0 otherwise. The objective function of the SDP system can now be expressed as $\mathcal{E} = \operatorname{tr}(FX)$, where $F = (F_{uv})$ is upper-diagonal. A vector $q = (q_u) \in \mathbb{R}^n$ is called *G-stochastic* if it is nonnegative and forms a valid probability distribution over each V_i (i.e., $\sum_{u \in V_i} q_u = 1$). Randomized rounding with respect to q produces an expected energy of $\operatorname{tr}(Fqq^T)$, and so, the drift is $\Delta = \operatorname{tr} F(qq^T - X)$. The problem, of course, is to find a suitable vector q. The next lemma provides a convenient criterion to test whether a given q provides a valid distribution.

Lemma 3.2 *Any nonnegative vector with L^1-norm p in the image space of X is G-stochastic.*

Proof: Recall that $X = W^T W$, where W is the matrix of column vectors (x_u). Let $\mathbf{1}_i$ be the 0/1 characteristic vector of V_i. Assuming that $q = Xy$ for some y, then

$$\sum_{u \in V_i} q_u = \mathbf{1}_i^T q = \mathbf{1}_i^T (W^T W) y = (W \mathbf{1}_i)^T W y = x_{u_0}^T W y,$$

where $W\mathbf{1}_i = x_{u_0}$ by Lemma 3.1. $x_{u_0}^T W y$ is independent of i and since by assumption $\|q\|_1 = p$, $\sum_{u \in V_i} q_u = 1$ for any i. □

By the Perron-Frobenius theorem for nonnegative matrices [45], the unit eigenvector z_1 corresponding to the largest eigenvalue λ_1 of X is nonnegative. We approximate X by

$$\widehat{X} = \lambda_1 z_1 z_1^T. \tag{7}$$

Let $s = (p/\|z_1\|_1)$ be the factor needed to scale z_1 to length p in the L^1 norm. Since z_1 is in the image space of X, the vector $q = sz_1$ is G-stochastic by Lemma 3.2. Perron-Frobenius rounding refers to the standard rounding rule applied now with respect to the distribution q. That is,

- PERRON-FROBENIUS ROUNDING RULE: For each $1 \le i \le p$, choose $u \in V_i$ at random with probability given by the u-th entry of z_1 scaled by s.

We can express the drift under this rounding scheme as

$$\begin{aligned}\Delta &= \operatorname{tr} F(qq^T - X) = \operatorname{tr} F\Big(s^2 z_1 z_1^T - X\Big) \\ &= \frac{s^2}{\lambda_1} \operatorname{tr} F\Big(\lambda_1 z_1 z_1^T - X\Big) + \Big(\frac{s^2}{\lambda_1} - 1\Big) \operatorname{tr} FX, \end{aligned} \tag{8}$$

and upper bound it as follows:

Lemma 3.3 *Let* $\mathbf{1}$ *denote the column vector of* n *ones and* U *the* n*-by-*n *matrix of ones, and let* $\{z_k\}$ *be an orthonormal eigenbasis of* X *with* $\lambda_k \ge 0$ *the eigenvalue associated with* z_k*. Then,*

$$\Delta \le (1+\delta)\|F\|_2 \sqrt{tr\, X^2 - \lambda_1^2} + \delta\, tr\, FX\,,$$

where $F = (F_{uv})$ *is the energy matrix and* X *is the solution matrix returned by the SDP system and*

$$\delta = \frac{tr\, UX}{tr\, U\widehat{X}} - 1 = \frac{\sum_{k>1} \lambda_k (\mathbf{1}^T z_k)^2}{p^2 - \sum_{k>1} (\mathbf{1}^T z_k)^2}.$$

Proof: We have $\delta = \Big(s^2/\lambda_1 - 1\Big)$ because

$$\begin{aligned}&\operatorname{tr} UX = \sum x_u^T x_v = \Big(\sum x_u\Big)^T \Big(\sum x_v\Big) = p^2 \|x_{u_0}\|_2^2 = p^2, \quad \text{and} \\ &\operatorname{tr} U\widehat{X} = \lambda_1 (\mathbf{1}^T z_1)^2 = \lambda_1 \|z_1\|_1^2;\end{aligned}$$

where the first follows because $\sum_{u \in V_i} x_u = x_{u_0}$ and there are p such positions i, and the second follows from the construction of $\widehat{X}$. Substituting δ into (8) and applying Cauchy-Schwarz, gives

$$\Delta \le (1+\delta)\|F\|_2\, \|X - \widehat{X}\|_2 + \delta \operatorname{tr} FX\,.$$

Note the dependence on the L^2 distance between X and its approximation $\widehat{X}$. We can express this distance in terms of the spectral weight placed on eigenvectors z_k for $k > 1$. The diagonalization of the matrix X gives the decomposition $X = \sum_k \lambda_k z_k z_k^T$; therefore, since

$X - \widehat{X}$ is symmetric and the z_k's are orthonormal,

$$\begin{aligned}\|X - \widehat{X}\|_2^2 = \operatorname{tr}(X - \widehat{X})^2 &= \operatorname{tr}\left(\sum_{k>1} \lambda_k z_k z_k^T\right)^2 \\ &= \sum_{k>1} \lambda_k^2 \operatorname{tr}(z_k z_k^T)^2 + 2 \sum_{k>l>1} \lambda_k \lambda_l \operatorname{tr}(z_k z_k^T)(z_l z_l^T) \\ &= \sum_{k>1} \lambda_k^2 = \operatorname{tr} X^2 - \lambda_1^2.\end{aligned}$$

Finally, using $p^2 - \operatorname{tr} UX = 0$, we have

$$\delta \equiv \frac{\operatorname{tr} UX}{\operatorname{tr} U\widehat{X}} - 1 = \frac{\operatorname{tr} U(X - \widehat{X})}{p^2 - \operatorname{tr} U(X - \widehat{X})} = \frac{\sum_{k>1} \lambda_k (\mathbf{1}^T z_k)^2}{p^2 - \sum_{k>1} (\mathbf{1}^T z_k)^2}.$$

□

Note that δ is quite small if the largest eigenvalue λ_1 carries most of the spectrum or z_1 is close to the vector $\mathbf{1}$, which makes the terms $\mathbf{1}^T z_k$ small for $k > 1$. The empirical results in Section 4 suggest that λ_1 may be much larger than the other eigenvalues in realistic situations.

The Uniform Case Projection rounding is expected to do well when the solution concentrates weight on a single node per position. What if the weights are nearly uniformly distributed? We can use Lemma 3.3 to argue that in this case even the strategy of uniform guessing has low drift.

Assume that (i) $x_{uu} = 1/|V_i|$ for all u, (ii) $x_{uv} = 1/|V_i||V_j|$ for any $(u, v) \in V_i \times V_j$ $(i < j)$, and (iii) $x_{uv} = 0$ for $u, v \in V_i$. (Note that although these assumptions are themselves unrealistic, the robustness of our arguments below makes them representative of the "uniform" end of the spectrum.) It is easy to construct an orthogonal eigenbasis for X:

Lemma 3.4 *The largest eigenvalue λ_1 of X is equal to $\sum_{i=1}^{p} |V_i|^{-1}$ and corresponds to the eigenvector*

$$z_1 = \frac{1}{\sqrt{\sum_i |V_i|^{-1}}} \sum_{i=1}^{p} |V_i|^{-1} \mathbf{1}_i \,,$$

where $\mathbf{1}_i$ is the 0/1 characteristic vector of V_i. None of the $n-1$ other eigenvectors are nonnegative: $p-1$ of them are of the form $\mathbf{1}_1 - \mathbf{1}_i$ and span the kernel of X, while, for each i, $|V_i| - 1$ of them are associated with the eigenvalue $|V_i|^{-1}$.

We defer the proof of Lemma 3.4 to the appendix.

Assume now that all the V_i's are of equal size n/p. Then $z_1 = (1/\sqrt{n})\mathbf{1}$ and Perron-Frobenius rounding degenerates into choosing solutions uniformly at random. Since all other eigenvectors are normal to z_1, by Lemma 3.3, we know that $\delta = 0$. Also, by Lemma 3.4,

$$\operatorname{tr} X^2 - \lambda_1^2 = \sum_{k>1} \lambda_k^2 = (n - p)p^2/n^2 \,.$$

If each F_{uv} is 0/1 and each node is connected to $2d$ neighbors, then $\|F\|_2 = \sqrt{nd}$ and, by Lemma 3.3, $\Delta \leq p\sqrt{d}$. This shows that, measured against the energy of $(p/n)^2 dn = dp^2/n$ of the random solution, the relative drift is at most $(n/p)/\sqrt{d}$, which is typically much less than 1 since each rotamer typically interacts with many rotamers in several positions.

Hence, if the solution matrix is sharply concentrated, we have shown projection rounding is expected to work well, at least under the assumption that the edge weights are nonnegative. In the opposite situation of a uniform solution matrix we have shown that for an unweighted, regular graph the drift is small if solutions are chosen uniformly at random.

4. Computational Results

We used the SDP formulation to design computationally the cores (i.e., the solvent inaccessible portions) of two proteins. Both proteins have a core β-barrel, a region where the backbone wraps around to form a structure reminiscent of the slats of a wooden barrel.

Our computational work focuses on protein cores because (1) this is where most pairwise interactions occur, (2) the cores are small, making them tractable for SDP, and (3) the energetics most important to the core residues are easier to model than those of solvent-exposed residues. In particular, since we are focusing on hydrophobic core interactions, we use an energy function that focuses on obtaining well-packed structures. More specifically, the interaction energies between rotamers (that is, the edge weights of our graph) are calculated using the 6–12 Lennard-Jones approximation to the van der Waal's force. Self-energies are calculated as the sum of the van der Waal's interaction between the rotamer and the backbone, plus a statistical term derived from the empirical probabilities listed in the rotamer library. Interactions between the side-chain and the backbone of flanking positions are ignored to account for some backbone flexibility. The statistical energy term for rotamer u is computed as $-\ln(p_u/p_0)$, where p_u is the probability of seeing rotamer u and p_0 is the probability of seeing the most common rotamer for that amino acid [10]. For all calculations, atom radii and interaction parameters are taken from AMBER96 [11], a commonly used package for evaluating the energy of protein conformations, with the radii of hydrogens reduced by 50% because of their uncertain position. The BALL C++ library [28] was used to manipulate the rotamers.

The ultimate test of protein design is to make the protein and confirm the predicted structure. Obviously, experimental work is beyond the scope of this paper, but the solutions to the design problems can be at least initially evaluated in several ways. First, we may expect the designed sequence to be similar to the native sequence since evolution has likely chosen a favorable sequence. (Note this need not be the case all the time; in fact, novel protein sequences that are considerably more stable than native protein sequences have been designed using fixed backbone approaches [37].) Second, since we are using an energy

function that focuses on packing, we expect the designed structure to avoid clashes between atoms and to pack the available space tightly. Third, we want the rounded solution to have energy near the optimum solution. We show below that our computationally designed cores generally fulfill these criteria.

In order to investigate the performance of the rounding schemes in a more controlled setting, we also experimented with two types of random graphs. We first consider uniform random graphs and then consider a family of randomly generated graphs that better model the interaction graphs observed in proteins.

The semidefinite programs were solved using version 6.0 of the SDPA [21] package, an implementation of an infeasible primal-dual interior-point method. The linear programs were solved using the dual simplex method with AMPL [19] and CPLEX 7.1 [26]. The SDP solutions were rounded using the projection and Perron-Frobenius methods described above.

We compare our SDP with the LP obtained by relaxing the integrality constraints from (4). The LP solutions were rounded by choosing node u with probability x_{uu}. For problems of this size, optimal integral solutions (denoted by OPT) can be found using formulation (4) and the integer programming option of CPLEX. This allows us to compute the relative gap of each found solution, as computed by $(x - OPT)/|OPT|$, where x is the value of the solution. For both the protein design problems and the simulated data, the SDP rounding schemes perform well, with significantly better average relative gaps.

4.1 Cold shock protein

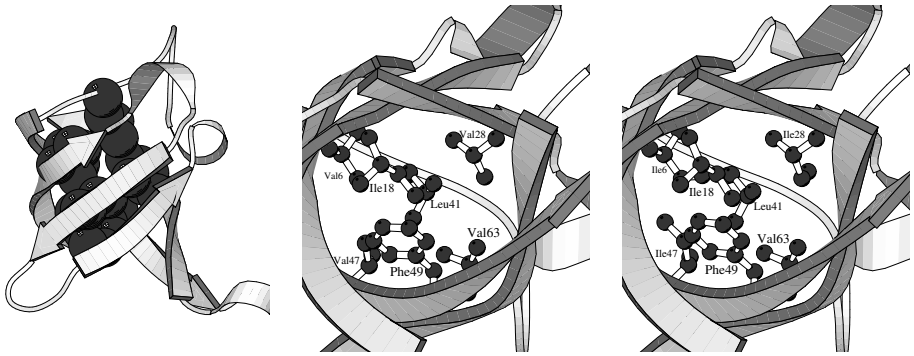
We applied the SDP method to the problem of redesigning the core of the *Bacillus caldolyticus* cold shock protein [38] (PDB code: 1c9o). Core residues were defined as having less than 1% of their surface area exposed to solvent, and determined by the program Surfv [40]. The following eight residues were found:

Val6, Gly16, Ile18, Val28, Leu41, Val47, Phe49, Val63.

The hydrophobic core positions (i.e., all positions listed above except position 16 with Gly) were varied, and allowed to assume any rotamer of the hydrophobic amino acids Ala, Val, Ile, Leu, Met, and Phe that occurred in the backbone-dependent rotamer library [16]. This yields 55 rotamers per position. These variable positions are shown in Figure 2.

The resulting problem had 385 nodes, 7 positions, and 63,313 nonzero cost matrix entries. Simple pairwise Goldstein DEE [23], a polynomial-time rule for throwing out rotamers that cannot possibly be in the optimal solution, was applied to the problem until no more nodes could be eliminated. This reduced the problem to 137 nodes, 7 positions, and 7,865 nonzero cost matrix entries.

The LP solution was rounded 1,000 times using the simple LP rounding scheme. The SDP solution was rounded 1,000 times with both the projection and Perron-Frobenius rounding



(a) (b) (c)

Figure 2: Cold-shock protein (`1c9o`). (a) The full protein, with the variable core atoms shown as black spheres and the axis of the β-barrel vertical; (b) the positioning of the side-chains in nature; (c) the solution returned by the SDP rounding schemes (the optimal). In (b) and (c) the protein is rotated so that we are looking down the axis of the barrel.

schemes. The minimum energy solution found is a good measure of how well one would do in practice, but this minimum energy may be influenced by the moderate search space size. The average energy of a rounded solution is a better indicator of the distribution obtained from rounding the relaxations. The best value over 1,000 roundings and the empirical average objective value in the limit are:

Method	**Best**	**Average**
LP	-217.2880	4058.6651
Projection	-238.4218	-102.2822
Perron-Frobenius	-238.4218	-209.3617

The optimum solution (determined by the integer programming option of CPLEX) is -238.4218. Both SDP rounding schemes find the optimum solution; this appears to correspond to a well-packed and plausible structure, as shown in Figure 2c. The average energy of the rounded solutions suggest that, as expected, the LP rounding scheme is a poor one. In fact, the average relative gap of the solutions found by the LP rounding scheme is 18.02, versus an average relative gap of 0.57 for projection rounding and 0.12 for Perron-Frobenius rounding. We expected the Perron-Frobenius rounding scheme to perform well, as the solution returned by the SDP has most its spectral weight placed on the largest eigenvalue (7.725 versus less than 0.05 for all the other eigenvalues).

The optimal choice has 57% sequence identity with the native sequence. Additionally, Figures 2b and 2c shows that the designed sequence packs more atoms into the core of the protein than the native structure. This is one indication that this sequence might be a good fit for this backbone as more tightly packed cores tend to be favored.

4.2 Triose phosphate isomerase

We applied the same procedure to the protein triose phosphate isomerase from chicken muscle [6] (PDB code: 1tim). This protein is an α/β-barrel, where the β-barrel core is surrounded by α-helical structures. We focused on the computational redesign of residues in the core of the β-barrel, as identified by Lesk [35], and shown in Figure 3. The 9 non-Glycine core residues are:

Val40, Ala62, Trp90, Ile92, Ile124, Val161, Ala163, Ile207, Leu230.

Trp90 was allowed to assume any rotamer of the aromatic amino acids Phe, His, Trp, and Tyr. The other residues were allowed to assume any rotamer of the hydrophobic amino acids Ala, Val, Ile, Leu, Met, and Phe. The same energy function was used as above. This resulted in 467 nodes and 91,737 nonzero edges. As with the cold shock protein, DEE was performed to throw out rotamers that cannot possibly be in the optimal solution; this reduced the problem to 141 nodes and 8,264 edges.

The optimal solution has objective value -208.5702. In this case, all methods find the optimal solution (shown in Figure 3c) within 1,000 roundings. The average objective values are:

Method	**Average**
LP	251.0156
Projection	93.1529
Perron-Frobenius	-36.9177

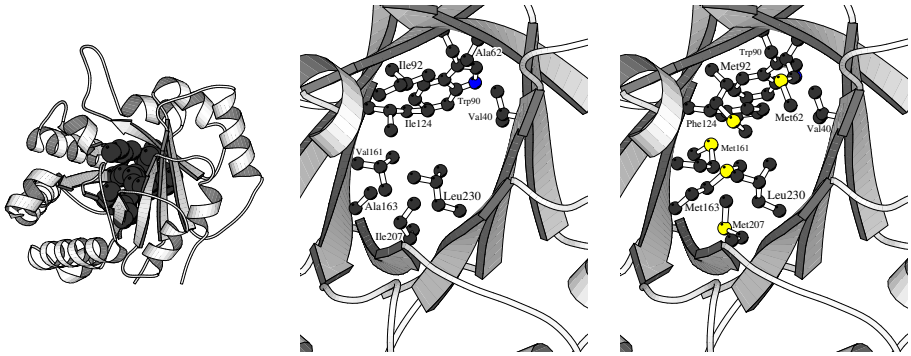
The average energy of the rounded solutions demonstrates that the Perron-Frobenius rounding again performs best for this problem. In fact, the SDP solution returned is close to a rank 1 matrix: the largest eigenvalue is 8.7563 out of a total spectral weight of 10; the second largest is 0.375. The average relative gap of the solutions found by Perron-Frobenius rounding is 0.82, versus 1.44 for for projection rounding and 2.20 for the LP rounding scheme.

The optimal solution avoids clashes, and, as can be seen from Figures 3a and 3b packs the available space well. It has only 33% sequence identity with the native solution. This is not necessarily unexpected: Mayo [12] gives a fold for which a sequence with only 21% identity to the native sequence that folds to the same shape.

4.3 Uniform Random Graphs

We consider the random graphs $G_U(n, p, r)$ parameterized by the number of nodes n, number of positions p, and edge probability r. Each position contains n/p nodes. Two nodes in different positions are connected by an edge with probability r. Each chosen edge is weighted by drawing a weight uniformly from $[0, 1]$. There are no self-edges.

We solved 30 instances of uniform random graphs with 60 nodes, 15 positions, and edge probability 0.5 using SDP. Figure 4a compares the fractional objective of the semidefinite



(a) (b) (c)

Figure 3: Triose phosphate isomerase (`1tim`). (a) The full protein, with the core atoms shown as black spheres; (b) the positioning of the side-chains in nature; (c) the solution returned by the SDP rounding schemes (the optimal). In (b) and (c) the protein is rotated so that we are looking down the axis of the barrel.

program with that of the linear program. The SDP provides a tighter lower bound on the minimum energy, typically within 10% of the optimum. In contrast, the fractional LP solution is never within 60% of the optimum. As expected then, as a side benefit, SDP provides a more effective bounding function than LP for branch-and-bound frameworks.

Figure 4b shows the best rounded solution found over 10,000 roundings. For these 30 graphs, both semidefinite rounding schemes outperform the LP in all cases, generally finding a solution within 10% of the optimum and only once finding a solution more than 20% above the optimum. This means, in a practical sense, that SDP allows us to find lower energy conformations. The two semidefinite rounding schemes are comparable, though Perron-Frobenius finds a lower energy solution in 11 out of 30 instances. In one case, projection rounding finds a better solution. The average rounded energy is shown in Figure 4c. Perron-Frobenius gives a slightly better distribution than projection rounding. Both SDP rounding schemes again outperform the LP one.

Figure 5 shows the 25 largest eigenvalues in descending order for each of the uniform random graphs shown in Figure 4. Most of the spectrum is concentrated in the first few eigenvalues, and so one would expect the $\widehat{X}$ in (7) to closely approximate the solution X and thus that Perron-Frobenius rounding would perform well.

These results remain qualitatively the same for other values of $p \geq 10$ and edge probabilities ≥ 0.3. For very sparse graphs, the SDP and LP methods yield similar results.

4.4 Neighborhood Random Graphs

The uniform random graphs do not capture several properties of real protein interaction graphs. Side-chains that are far apart in the folded protein structure typically do not interact.

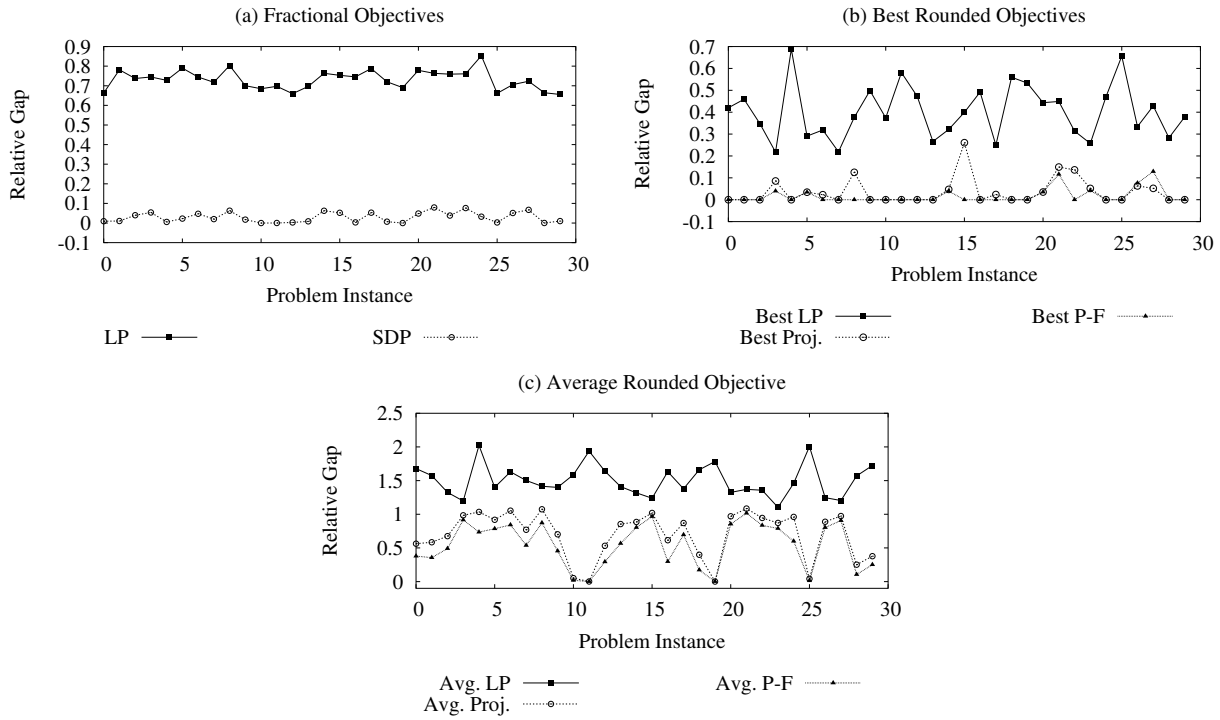


Figure 4: Uniform random graphs with 60 nodes, 15 positions, and edge probability 0.5. Edge weights were drawn uniformly from $[0, 1]$. Results for 30 random instances are shown. The relative gap of an objective value x (fractional or rounded) is defined to be $|(x - OPT)/OPT|$.

On the other hand if two residues are near each other in the folded structure most of their rotamers will interact.

We consider neighborhood random graphs $G_N(n, p, d)$ that capture some of these properties. They are again parameterized by the number of nodes n and number of positions p, where each position has n/p nodes. Given parameter d, edges are defined as follows: for each position j a point b_j is chosen uniformly at random in the 3D unit cube. If the Euclidean distance between b_i and b_j is $\leq d$, then the rotamers in positions i and j are connected by the complete bipartite graph; if the distance is $> d$, there are no edges between i and j. Edges are weighted by choosing a weight uniformly from $[0, 1]$.

Figure 6 shows the results for neighborhood random graphs with various values for d. For sparse graphs (small d), the SDP and LP approaches yield similar results. As d increases, positions are more likely to be connected, the optimum objective grows, and the SDP's advantage in lower bounding the optimum solution increases. Both projection rounding and Perron-Frobenius rounding can find the optimum solutions for most of these graphs within 10,000 roundings, whereas this is not the case for the LP rounding. The average rounded energy is shown in Figure 6, and again Perron-Frobenius gives a slightly better distribution than projection rounding. The spectra for neighborhood random graphs of low connection

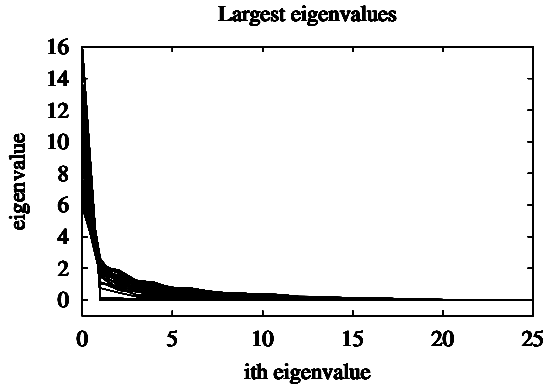


Figure 5: The 25 largest eigenvalues of the SDP solution matrix for the 30 uniform random graphs shown in Figure 4. The eigenvalues sum to 16 because $\mathrm{tr}\, X = p$ and there are 16 positions including the dummy position V_0 in (3).

distance are also very concentrated in the highest eigenvalue. As the connection distance increases, the spectrum generally becomes more spread out (data not shown).

5. Inapproximability

While some NP-complete problems permit approximation algorithms (i.e., algorithms that guarantee that all their solutions are within some factor of optimal), here we show that this is not the case for the side-chain positioning problem. That is, it is even hard to compute any reasonable approximation of the minimum energy with a theoretical guarantee.

The SCP problem is an optimization problem, not a language membership problem. It is turned into the latter by providing as input both an instance of a side-chain positioning problem and an integer k, and asking whether the GMEC has energy less than k. The statements below are to be understood in that context.

Theorem 5.1 *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 number of residue positions.*

If the minimum energy is close to 0, of course it might not be too surprising to hear that a good multiplicative approximation is hard to find. The strength of our result is that it is still very hard to find even if the minimum energy is bounded away from 0 by a constant.

Proof: A 3-CNF formula is a conjunction of clauses, each one consisting of the disjunction of three literals (not necessarily distinct). The PCP theorem [4, 5] 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. Furthermore, if 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

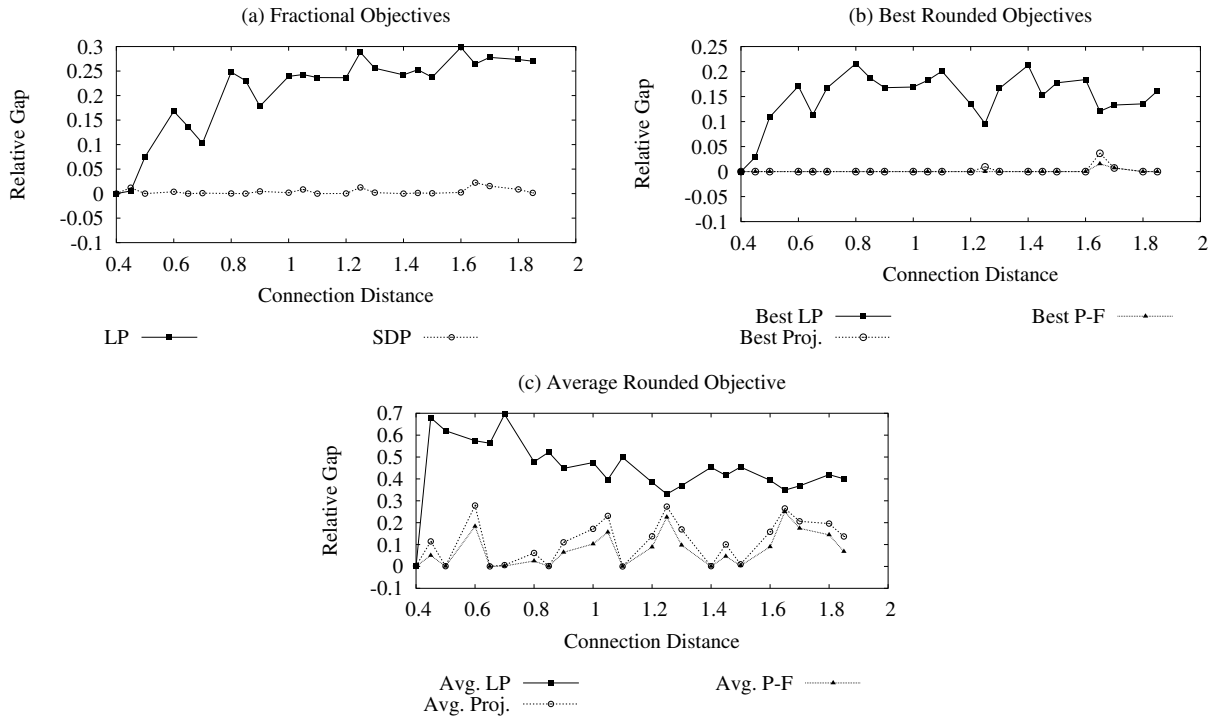


Figure 6: Neighborhood random graphs with 60 nodes, 15 positions for various connection distances d. Edge weights were drawn uniformly from $[0, 1]$.

polynomial time. This implies that it is NP-hard to distinguish between formulas that are satisfiable and those that are strongly unsatisfiable.

Given a 3-CNF formula with p clauses, we create an SCP problem such that if the formula is satisfiable then GMEC $= 0$, but if the formula is not satisfiable then the GMEC will tell us how many clauses can be satisfied. Thus, the discussion above will imply that there cannot be a polynomial time algorithm that always computes even a reasonable approximation to the GMEC (unless P=NP).

We build a p-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. 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 edge is assigned weight 3. The 4th vertex in each V_i is an "extra" vertex with no adjacent edges and vertex weight 1.

If the CNF formula is satisfiable then, for each V_i we select a literal set to true as the GMEC vertex. Obviously, these p vertices form an independent set (since one cannot set both a variable and its negation to true) and the energy of the system is zero.

If the CNF formula is not satisfiable then the GMEC is formed by picking the largest independent set among the vertices that correspond to clauses, including at most one vertex per V_i, and completing the selection with extra vertices. (Picking any pair of adjacent vertices

would be a mistake since that choice could be locally improved.) We can set to true the literals corresponding to the vertices of the independent set. Therefore, the energy of the GMEC is $p - c$, where c is the maximum number of satisfiable clauses in the CNF formula. Thus, it is NP-hard to tell apart a side-chain positioning problem with minimum energy 0 and one with minimum energy $(1 - \alpha)p = (1 - \alpha)n/4$.

For a more realistic scenario where the minimum energy is bounded away from 0, we can add an extra V_i consisting of a single unit-weight vertex; then, we still cannot hope to find an approximation that is better than a factor $\Omega(n)$ of the minimum energy. □

6. Conclusions

We formulate the side-chain positioning problem as an instance of semidefinite programming and introduce two new rounding schemes for converting fractional solutions into integral ones. Our rounding schemes appear quite general and we hope they can be used elsewhere.

We have applied our method to the problem of computationally redesigning the cores of two naturally occurring proteins. In addition, we have investigated how the rounding formulations behave on two classes of random graphs. While the hardness of the SCP problem argues that no method will do well in general, our computational experiments confirm the effectiveness of our methods. We provide a measure of theoretical justification for this.

We have shown semidefinite programming can be applied to biological problems of realistic, albeit small, size. Though non-polynomial search heuristics are more practical at present for larger biological problems, as semidefinite programming algorithms and solvers improve, our approach will become more attractive. Interesting directions for future work include finding a faster SDP algorithm specialized to our system of constraints, as well as developing better rounding schemes that better exploit the underlying real-world statistical properties of the problem.

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5(1):13–51, 1995.

[2] N. Alon and N. Kahale. Approximating the independence number via the θ-function. *Math. Programming*, 80:253–264, 1998.

[3] 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.

[4] 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.

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

[6] D. W. Banner, A. Bloomer, G. A. Petsko, D. C. Phillips, and I. A. Wilson. Atomic coordinates for triose phosphate isomerase from chicken muscle. *Biochem. Biophys. Res. Commun.*, 72(1):146–55, Sept. 1976.

[7] S. J. Benson, Y. Ye, and X. Zhang. Mixed linear and semidefinite programming for combinatorial and quadratic optimization. *Optimization Methods and Software*, 11:515–544, 1999.

[8] D. Bertsimas and Y. Ye. Semidefinite relaxations, multivariate normal distributions, and order statistics. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 3, pages 1–19. Kluwer Academic Publishers, 1998.

[9] R. B. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, pages 280–285, 1987.

[10] M. J. Bower, F. E. Cohen, and R. 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.

[11] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, Jr., D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Am. Chem. Soc.*, 117:5179–5197, 1995.

[12] B. I. Dahiyat and S. L. Mayo. De novo protein design: Fully automated sequence selection. *Science*, 278:82–87, Oct. 1997.

[13] J. Desmet, M. De Maeyer, and I. Lasters. The "dead end elimination" theorem as a new approach to the side-chain packing problem. In K. Merz and S. LeGrand, editors, *The Protein Folding Problem and Tertiary Structure Prediction*, pages 307–337. Birkhäuser Boston, Inc., 1994.

[14] J. Desmet, M. D. Maeyer, B. Hazes, and I. Lasters. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356:539–542, Apr. 1992.

[15] W. E. Donath and A. Hoffman. Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices. *IBM Technical Disclosure Bulletin*, 15:938–944, 1972.

[16] R. L. Dunbrack Jr and M. Karplus. Backbone-dependent rotamer library for proteins: application to side-chain prediction. *J. Mol. Biol.*, 230:543–574, 1993.

[17] O. Eriksson, Y. Zhou, and A. Elofsson. Side chain-positioning as an integer programming problem. In *Proceedings of 1st Workshop on Algorithms in BioInformatics*, BRICS, University of Aarhus, Denmark, Aug. 2001.

[18] U. Feige and J. Kilian. Heuristics for finding large independent sets, with applications to coloring semi-random graphs. In *Proceedings of the 39th Annual IEEE Symp. Found. Comput. Sci.*, pages 674–683, Nov. 1998.

[19] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming.* Duxbury Press / Brooks/Cole Publishing Company, 2002.

[20] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18(1):61–77, 1997.

[21] K. Fujisawa, M. Fukuda, M. Kojima, and K. Nakata. Numerical evaluation of SDPA. Technical Report B-330, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152, Sept. 1997.

[22] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995.

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

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

[25] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, 2nd edition, 1994.

[26] ILOG CPLEX 7.1 `http://www.cplex.com/`.

[27] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, Mar. 1998.

[28] O. Kohlbacher and H.-P. Lenhof. Ball — rapid software prototyping in computational molecular biology. *Bioinformatics*, 16(9):815–824, 2000.

[29] 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.

[30] H. C. Lau. A new approach for weighted constraint satisfaction. *Constraints*, 7:151–165, 2002.

[31] H. C. Lau and O. Watanabe. Randomized approximation of the constraint satisfaction problem. In R. Karlsson and A. Lingas, editors, *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory*, pages 76–87, 1996.

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

[33] C. Lee. Predicting protein mutant energetics by self-consistent ensemble optimization. *J. Mol. Biol.*, 236(3):918–939, Feb. 1994.

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

[35] A. M. Lesk, C.-I. Brändén, and C. Chothia. Structural principles of α/β barrel proteins: The packing of the interior of the sheet. *Proteins*, 5:139–148, 1989.

[36] L. Lovász. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25:1–7, 1979.

[37] 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.

[38] U. Mueller, D. Perl, F. X. Schmid, and U. Heinemann. Thermal stability and atomic-resolution crystal structure of the Bacillus caldolyticus cold shock protein. *J. Mol. Biol.*, 297(4):975–988, Apr. 2000.

[39] Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms*. SIAM, Philadelphia, 1993.

[40] A. Nicholls, K. A. Sharp, and B. Honig. Protein folding and association: Insights from the interfacial and thermodynamic properties of hydrocarbons. *Proteins*, 11(4):281–296, 1991.

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

[42] J. W. Ponder and F. M. Richards. Tertiary templates for proteins. use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.*, 193(4):775–791, Feb. 1987.

[43] P. Raghavan and C. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

[44] J. D. P. Rolim and L. Trevisan. A case study of de-randomization methods for combinatorial approximation problems. *Journal of Combinatorial Optimization*, 2(3):219–236, 1998.

[45] E. Seneta. *Non-negative Matrices and Markov Chains.* Springer-Verlag, New York, 2nd edition, 1981.

[46] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, Mar. 1996.

[47] U. Zwick. Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 679–687, 1999.

A. Appendix

Lemma 3.4 *The largest eigenvalue λ_1 of X is equal to $\sum_{i=1}^{p} |V_i|^{-1}$ and corresponds to the eigenvector*

$$z_1 = \frac{1}{\sqrt{\sum_i |V_i|^{-1}}} \sum_{i=1}^{p} |V_i|^{-1} \mathbf{1}_i \,,$$

where $\mathbf{1}_i$ is the 0/1 characteristic vector of V_i. None of the $n-1$ other eigenvectors are nonnegative: $p-1$ of them are of the form $\mathbf{1}_1 - \mathbf{1}_i$ and span the kernel of X, while, for each i, $|V_i|-1$ of them are associated with the eigenvalue $|V_i|^{-1}$.

Proof: If $\widehat{X} = \lambda_1 z_1 z_1^T$, then $X - \widehat{X}$ is the n-by-n matrix made of blocks $B_1, \ldots, B_p$ along the diagonals and 0 everywhere: each B_i is a $|V_i|$-by-$|V_i|$ circulant matrix with $|V_i|^{-1} - |V_i|^{-2}$ along the diagonal and $-|V_i|^{-2}$ elsewhere. The eigenvectors of an m-by-m circulant matrix consist of the rows of the matrix of the Fourier transform over the additive group $\mathbf{Z}/m\mathbf{Z}$: For $k > 0$, this gives us an eigenvector $(1, e^{2\pi ik/m}, \ldots, e^{2\pi ik(m-1)/m})$ for each $0 < k < m$. The corresponding eigenvalue for B_i is $|V_i|^{-1}$ (hence, both its algebraic and geometric multiplicities are $|V_i| - 1$). The corresponding eigenvectors of X are derived trivially by padding with zeroes at the appropriate places. Note that we must skip the case $k = 0$,

because the eigenvector of B_i that gets padded into $\mathbf{1}_i$ is *not* an eigenvector of X. To complete the diagonalization of X, we must resolve its kernel. Going back to the relations $x_{uu} = \sum_{v \in V_j} x_{uv}$, we easily verify that $\operatorname{Ker} X$ is spanned by the $p-1$ vectors $\mathbf{1}_1 - \mathbf{1}_i$, for $1 < i \leq p$. $\square$