The Protein Sequence Design Problem in Canonical Model on 2D and 3D Lattices

Piotr Berman[1], Bhaskar DasGupta[2], Dhruv Mubayi[3], Robert Sloan[2], György Turán[3], and Yi Zhang[2]

[1] Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802. Email: berman@cse.psu.edu
[2] Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607-7053. Email: {dasgupta,sloan,yzhang3}@cs.uic.edu
[3] Department of Mathematics, Statistics & Computer Science, University of Illinois at Chicago, Chicago, IL 60607-7045. Email: mubayi@math.uic.edu and gyt@uic.edu

Abstract. In this paper we investigate the **protein sequence design (PSD)** problem (also known as the **inverse protein folding** problem) under the **Canonical model** [4] **on 2D and 3D lattices** [12, 25]. The Canonical model is specified by **(i)** a *geometric representation* of a target protein structure with amino acid residues via its *contact graph*, **(ii)** a *binary folding code* in which the amino acids are classified as *hydrophobic* (H) or *polar* (P), **(iii)** an *energy function* Φ defined in terms of the target structure that should *favor* sequences with a *dense hydrophobic core* and *penalize* those with *many solvent-exposed hydrophobic residues* (in the Canonical model, the energy function Φ gives an H-H residue contact in the contact graph a value of -1 and all other contacts a value of 0), and **(iv)** to prevent the solution from being a biologically meaningless all H sequence, the number of H residues in the sequence S is limited by fixing an upper bound λ on the ratio between H and P amino acids. The sequence S is designed by specifying which residues are H and which ones are P in a way that realizes the *global minima* of the energy function Φ. In this paper, we prove the following results:

(1) An earlier proof of NP-completeness of finding the global energy minima for the PSD problem on 3D lattices in [12] was based on the NP-completeness of the same problem on 2D lattices. However, the reduction was not correct and we show that the problem of finding the global energy minima for the PSD problem for 2D lattices can be solved *efficiently* in *polynomial time*. But, we show that the problem of finding the global energy minima for the PSD problem on 3D lattices is indeed NP-complete by a providing a different reduction from the problem of finding the largest clique on graphs.

(2) Even though the problem of finding the global energy minima on 3D lattices is NP-complete, we show that an *arbitrarily* close approximation to the global energy minima can indeed be found efficiently by taking

[4] The Canonical model is neither the same nor a subset of the Grand Canonical (GC) model in [19, 24]; see Section 1.3 for more details.

appropriate combinations of optimal global energy minima of substrings of the sequence S by providing a polynomial-time approximation scheme (PTAS). Our algorithmic technique to design such a PTAS for finding the global energy minima involves using the *shifted slice-and-dice approach* in [6, 17, 18]. This result improves the previous best polynomial-time approximation algorithm for finding the global energy minima in [12] with a performance ratio of $\frac{1}{2}$.

1 Introduction

In protein structure studies the single most important research problem is to understand how protein sequences fold into their native 3D structures, *e.g.*, see [3, 5, 7, 9, 12–16, 21, 22, 26, 27]. This problem can be investigated at two *complementary* levels. At a *lower* level, one wishes to determine how an individual protein sequence folds. The problem of using sequence input to generate 3D structure output is referred to as the *ab initio protein structure prediction* problem and has been shown to be NP-hard [3, 5, 7]. At a *higher* level, one wants to analyze the *protein landscapes*, *i.e.*, the relationship between the space of all protein sequences and the space of native 3D structures. A formal framework for analyzing protein landscapes is established by a model that relates a set S of protein sequences to a set P of protein structures. Typically this is given by a real-valued *energy* function $\Phi : S \times P \rightarrow \mathbb{R}$ that models the "fit" of a sequence $s \in S$ to a structure $p \in P$ according to the principles of statistical mechanics. A functional relationship between sequences and structures is obtained by *minimizing* Φ with respect to the structures, *i.e.*, a structure q *fits* a sequence s if $\Phi(s, q) = min_{p \in P} \Phi(s, p)$. Typically the values of Φ are assumed to model notions of free energy and the minimization is supposed to provide approximations to the *most probable structure* obtained from thermodynamical considerations.

The exact nature of Φ depends on the particular model but, for any given specification, there is natural interest in the fine-scale structure of Φ. For example, one might ask whether a certain kind of protein structure is more likely to be the native structure of a diverse collection of sequences (thus making structure prediction from sequences difficult). One approach to investigating the structure of Φ is to solve what is called the *protein sequence design* (PSD) or the *inverse protein folding* problem: given a target 2D or 3D structure as input, return a *fittest* sequence with respect to Φ. Three criteria have been proposed for evaluation of the fitness of the protein sequence with respect to the target structure: **(a)** the sequence should fold to the target structure, **(b)** there should be *no degeneracy* in the ground state of the sequence and **(c)** there should be a *large gap* between the energy of the sequence in the target structure and the energy of the sequence in any other structure. Some researchers [27] have proposed weakening condition **(b)** by requiring that the degeneracy of the sequence be no greater than the degeneracy of any other sequence that also folds to the target structure. The PSD problem has been investigated in a number of studies [4, 8, 10, 12, 19, 23–25, 27]. The computational complexity of PSD in its full generality as described above is unknown but conjectured to be NP-hard; the currently best known algorithms are by exhaustive search or Monte Carlo simulations.

One possible mode of handling the PSD problem is by defining a *heuristic sequence design* (HSD) problem where a simplified pair-wise interaction function is used to compute the landscape energy function Φ. The implicit assumption is that a sequence that satisfies the HSD problem also solves PSD. Several quantitative models have been proposed for the HSD problem in the literature [8, 24, 25]. This paper is concerned with the Canonical model of Shahknovich and Gutin [25]. This model is specified by **(1)** a *geometric representation* of a target protein structure with n amino acid residues via its *contact graph*, **(2)** a *binary folding code* in which the amino acids are classified as *hydrophobic* (H) or *polar* (P) [9, 20], and **(3)** an *energy function* Φ defined in terms of the target structure that should *favor* sequences with a *dense hydrophobic core* and *penalize* those with *many solvent-exposed hydrophobic residues*. To design a sequence S, we must specify which residues are H and which ones are P. Thus, S is a sequence of n symbols each of which is either H or P. In the Canonical model, the energy function Φ gives a H-H residue contact in the contact graph a value of -1 and all other contacts a value of 0. To prevent the solution from being a biologically meaningless all H sequence, the number of H residues in S is limited by fixing an upper bound λ of the ratio between H and P amino acids. The Canonical model gives rise to the following *special case* of the *densest subgraph problem* on K vertices (denoted by the PSDC$_2$ and the PSDC$_3$ Problems):

Definition 1.
(a) *A* d-dimensional lattice *is a graph* $G(n,d) = (V(n,d), E(n,d))$ *with* $V(n,d) = \times_{i=1}^{d}\{-n, -n+1, \ldots, n-1, n\}$ *for some positive integer* n *and* $E(n,d) = \{\{(i_1, \cdots, i_d), (j_1, \cdots, j_d)\} \ : \ \sum_{k=1}^{d} |i_k - j_k| = 1\}$ *($X \times Y$ denote the Cartesian product of two sets X and Y).*

(b) *A* 2D sequence *(resp.* 3D sequence*)* $S = (V, E)$ *is a graph that is a simple path in* $G(n,2)$ *(resp.* $G(n,3)$*) for some n; the contact graph of such a 2D sequence (resp. 3D sequence) S is a graph* $\bar{G} = (\bar{V}, \bar{E})$ *where* $\bar{E}$ *consists of all edges* $\{u, v\} \in E(n,2)$ *(resp.* $\{u, v\} \in E(n,2)$*) such that* $u, v \in V$ *and* $\{u, v\} \notin E$ *and* $\bar{V}$ *is the set of end points of the edges in* $\bar{E}$*.*

*Problem 1 (*DS Problem). The Densest Subgraph (DS) problem has a graph $G = (V, E)$ and a positive integer K as inputs, and the goal is to find a $V' \subseteq V$ with $|V'| \leq K$ that *maximizes* $|\{(u, v) \in E \ : \ u, v \in V'\}|$.

*Problem 2 (*PSDC$_2$/PSDC$_3$ Problems). The PSD problem for the Canonical model on a 2D (resp. 3D) lattice, denoted by PSDC$_2$ (resp. PSDC$_3$), is an instance of the DS problem when the input graph G is the contact graph realized by a 2D (resp. 3D) sequence.

References [1, 2] consider the DS problem for general graphs. Hart [12] considers both PSDC$_2$ and PSDC$_3$ problems, provides approximation algorithm for PSDC$_3$ with an approximation ratio of $\frac{1}{2}$ and an *almost* optimal algorithm for

PSDC_2. The following property of the contact graph of a 2D/3D sequence is easy to observe [12]:

> the contact graph G for a 2D sequence (resp. 3D sequence) is a graph that is a subgraph of the 2D lattice (respectively, 3D lattice) with at most two vertices of degree 3 (resp. 5) and all other vertices of degree at most 2 (resp. 4).

1.1 Our Results

Throughout the rest of the paper, G is the given input graph in our problems, K is the maximum number of residues that can be hydrophobic and $V(H)$ (resp. $E(H)$) is the vertex set (resp. edge set) of any graph H. Our results are:

(I) We show that the problem of finding the global energy minima for the PSD problem for 2D lattices can be solved in polynomial time (see Section 2).

(II) We show that the problem of finding the global energy minima for the PSD problem on 3D lattices is NP-complete by showing that the PSDC_3 decision problem is NP-complete via a reduction from the problem of finding the largest clique on graphs (see Section 3.1). An earlier proof of NP-completeness of this problem in [12] was based on an incorrect proof of NP-completeness of the same problem on 2D lattices.

(III) Even though the problem of finding the global energy minima on 3D lattices is NP-complete, we show that an arbitrarily close approximation to the global energy minima can indeed be found efficiently by taking appropriate combinations of optimal global energy minima of substrings of the sequence S by providing a polynomial-time approximation scheme (PTAS) for the PSDC_3 problem (see Section 3.2). This result improves the previous best polynomial-time approximation algorithm for finding the global energy minima in [12] which had a performance ratio of $\frac{1}{2}$.

1.2 Summary of Algorithmic Techniques Used

- The polynomial-time algorithm in Result **(I)** uses the polynomial-time Generalized Knapsack problem, the special topology of the input contact graph as mentioned at the end of the introduction and the fact that the range of Φ are small integers.
- The NP-completeness reduction in Result **(II)** uses the NP-completeness reduction in [11] from the maximum clique problem to the densest subgraph problem on general graphs. The challenging and tedious parts in our reduction is to make sure that the reduction works for the special topology of our input contact graph and that such a contact graph can in fact be realized by a 3D sequence.
- The PTAS in Result **(III)** is designed using the *shifted slice-and-dice approach* in [6, 17, 18].

1.3 Difference Between the Canonical and the Grand Canonical Model

To avoid possible confusion due to similar names, we would like to point out that the Canonical model considered in this paper is neither the same nor a subset of the Grant Canonical (GC) model for the protein sequence design problem [19, 24]. The GC model is defined by a different choice of the energy function Φ. In particular, let S_H to denote the set of numbers i such that the $i^{\rm th}$ position in S is equal to H. Then, Φ is defined by the equation $\Phi(S) = \alpha \sum_{i,j \in S_H, i<j-2} g(d_{ij}) + \beta \sum_{i \in S_H} s_i$, where $\alpha < 0$, $\beta > 0$, s_i is the area of the solvent-accessible contact surface for the residue (in Å), d_{ij} is the distance between the residues i and j (in Å) and $g = \begin{cases} 1/[1+\exp(d_{ij}-6.5)] & \text{when } d_{ij} \leq 6.5 \\ 0 & \text{when } d_{ij} > 6.5 \end{cases}$ is a *sigmoidal* function. The scaling parameters α and β have default values -2 and $\frac{1}{3}$, respectively.

1.4 Basic Definitions and Notations

For two graphs G_1 and G_2, $G_1 \cup G_2$ denotes the graph with $V(G_1 \cup G_2) = V(G_1) \cup V(G_2)$ and $E(G_1 \cup G_2) = E(G_1) \cup E(G_2)$. H_S is the subgraph of H induced by the vertex set S, *i.e.*, $V(H_S) = S$ and $E(H_S) = \{(x,y) \in E(H) \mid x, y \in S\}$. $n_0(H), n_1(H)$ and $n_2(H)$ denote the number of vertices in the connected components of a graph H with zero, one or two cycles, respectively. $H \setminus S$ denotes the graph obtained from a graph H by removing the vertices in S and all the edges incident to these vertices in S. For a vertex (x,y,z) of the 3D lattice, x, y and z are the $1^{\rm st}$, $2^{\rm nd}$ and $3^{\rm rd}$ coordinate, respectively. $[i,j]$ and $[i,j)$ denote the set of integers $\{i, i+1, i+2, \ldots, j\}$ and $\{i, i+1, i+2, \ldots, j-1\}$, respectively. $\mathrm{OPT}(G,K)$ denotes the number of edges in an optimal solution to the PSDC_2 or PSDC_3 problem. A δ-approximate solution (or simply a δ-approximation) of a maximization problem is a solution with an objective value no smaller than δ times the value of the optimum; an algorithm of *performance* or *approximation ratio* δ produces an δ-approximate solution. A *polynomial-time approximation scheme* (PTAS) for a maximization problem is an algorithm that, for any given *constant* $\varepsilon > 0$, runs in polynomial time and produces an $(1-\varepsilon)$-approximate solution.

2 The PSDC_2 Problem

In [12] Hart provided a proof of NP-completeness of PSDC_2. Unfortunately, the proof was not correct because the reduction from the Knapsack problem was pseudo-polynomial time and Knapsack problem is not strongly NP-complete. We show in the following lemma that PSDC_2 can indeed be solved in polynomial time. Due to space limitations, we omit the proof of the following lemma.

Lemma 1. *There exists an $O(K|V(G)|)$ time algorithm that solves $PSDC_2$.*

3 The PSDC$_3$ Problem

In the first subsection, we show that the PSDC$_3$ problem is NP-complete even though the PSDC$_2$ problem is not. In the second subsection, we show how to design a PTAS for the PSDC$_3$ problem using the shifted slice-and-dice technique.

3.1 NP-completeness Result for PSDC$_3$

Theorem 1. *The PSDC$_3$ problem is NP-complete.*

Proof. It is trivial to see that PSDC$_3$ is in NP. To show NP-hardness, we provide a reduction from the CLIQUE problem on graphs whose goal is to decide, for a given graph G and an integer k, if there is a complete subgraph (clique) of G of k vertices. Let us denote by 3DS problem the DS problem on graphs with a maximum degree of 3. We will use a minor modification of a reduction of Feige and Seltser [11] from the CLIQUE problem to the the 3DS problem along with additional arguments. Consider an instance (G, k) of the CLIQUE problem where $V(G) = (v_1, \ldots, v_n)$ with $|V(G)| = n$. We can assume without loss of generality that n is an *exact* power of 2, n is sufficiently large and the vertex v_n has zero degree[5]. Let $t_1 \ll t_2 \ll t_3 \ll t_4 \ll t_5 \ll t_6$ be six sufficiently large polynomials in n; for example, $t_1 = n^{20}$ and $t_i = t_{i-1}^2$ for $i \in [2, 6]$ suffices. From G, we construct an instance graph H of the 3DS problem using a minor modification of the construction in Section 3 of Feige and Seltser [11] as follows:

- Replace each vertex v_i by a simple cycle of "cycle" edges
$$C^i = \{v_1^i, v_2^i\}, \{v_2^i, v_3^i\}, \ldots, \{v_{2nt_4-1}^i, v_{2nt_4}^i\}, \{v_{2nt_4}^i, v_1^i\} \in E(H)$$
on the $2nt_4$ new "cycle" vertices $v_1^i, v_2^i, \ldots, v_{2nt_4}^i \in V(H)$.
- Replace each edge $\{v_i, v_j\} \in E(G)$ with $i < j$ by a simple path of "path" edges
$$P^{ij} = \{\{v_{(n+j)t_4}^i, u_1^{ij}\}, \{u_1^{ij}, u_2^{ij}\} \ldots, \{u_{kt_5-1}^{ij}, u_{kt_5}^{ij}\}, \{u_{kt_5}^{ij}, v_{(n+i)t_4}^j\}\} \subseteq E(H)$$
of $kt_5 + 2 > 2nkt_4$ vertices between $v_{(n+j)t_4}^i$ and $v_{(n+i)t_4}^j$ where $u_1^{ij}, u_2^{ij}, \ldots, u_{kt_5}^{ij} \in V(H)$ are the new "path" vertices.
- Finally, we add a set of s additional separate connected components $Q_1, Q_2, \ldots, Q_s$, which will be specified later, such that all vertices in $\cup_{i=1}^s Q_i$ are of degree *at most* 2, no Q_i is an odd cycle and $\cup_{i=1}^s |V(Q_i)|$ is a polynomial in n.

Let $K = 2nkt_4 + \binom{k}{2} kt_5$ and $m = 2nkt_4 + \binom{k}{2}(kt_5 + 1)$. The same proof in Feige and Seltser [11] works to show that, for *any* selection of $Q_1, \ldots, Q_s$, there exists a subgraph with K vertices and at least m edges in H if and only if G has a clique of k vertices. Thus, to complete our reduction, we need to show the following:

[5] The degree assumption for v_n helps us to design the sequence $\mathcal{S}$ whose contact map will correspond to the graph H for the 3DS problem that we generate from an instance of the CLIQUE problem.

Step 1 (embedding H in the 3D lattice) H can be embedded in the 3D lattice.

Step 2 (realizing H as a contact graph) For some choice of $Q_1, Q_2, \ldots, Q_s$ H is the contact graph of a 3D sequence $\mathcal{S}$.

Details of both these steps are omitted due to space limitations.

Corollary 2 *3DS is NP-complete even if G is a subgraph of the 3D lattice.*

3.2 An Approximation Scheme via Shifted Slice-and-dice

All the graphs discussed in this section are subgraphs of the 3D lattice. For notational convenience and simplifications we assume, without loss of generality, that our input graph G satisfies $V(G) \subseteq \times_{i=1}^{3}[0, n_i)$ for some n_1, n_2, n_3 with $|V(G)| \geq \max\{n_1, n_2, n_3\}$. We classify an edge $\{(i_1, i_2, i_3), (j_1, j_2, j_3)\} \in E(G)$ as *horizontal*, *vertical* or *lateral* if $i_1 \neq j_1$, $i_2 \neq j_2$ or $i_3 \neq j_3$, respectively. Let E_{-}, $E_{|}$ and $E_{/}$ be the set of horizontal, vertical and lateral edges in an optimal solution.

Theorem 3. *For every $\varepsilon > 0$, there is an $O\left(\frac{K}{\varepsilon^3} 2^{1/\varepsilon^3} |V(G)|\right)$ time algorithm that returns a solution of the PSDC$_3$ problem with at least $(1-\varepsilon)OPT(G, K)$ edges.*

Proof. We use the shifted slice-and-dice technique of [6, 17, 18]. For convenience, we use the following notations:

- $\nu_j = \left\lfloor \frac{n_j - 1}{\ell} \right\rfloor$ for $j \in [1, 3]$,
- $\kappa_1 = [i\ell + \alpha, \min\{(i+1)\ell, n_1\} + \alpha)$ $\kappa_2 = [j\ell + \alpha, \min\{(j+1)\ell, n_2\} + \alpha)$ and $\kappa_3 = [k\ell + \alpha, \min\{(k+1)\ell, n_3\} + \alpha)$ for some specified values i, j, k and number α.

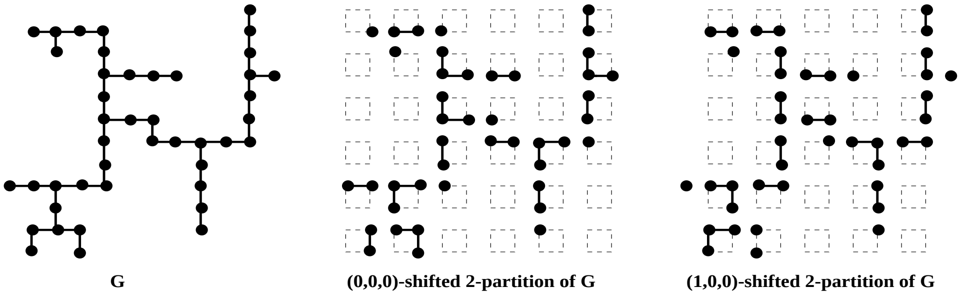
We first need the following definition.

Definition 2. *For a given positive integer (partition length) $\ell > 0$ and three positive integers (shifts) $0 \leq \alpha, \beta, \gamma < \ell$, an (α, β, γ)-shifted ℓ-partition of G, denoted by $\mathit{\Pi}_{\ell}^{\alpha,\beta,\gamma}[G]$ is the subgraph of G in which $V(\mathit{\Pi}_{\ell}^{\alpha,\beta,\gamma}[G]) = V(G)$ and $E(\mathit{\Pi}_{\ell}^{\alpha,\beta,\gamma}[G])$ is exactly*

$$E(G) \cap \left(\bigcup_{i=0}^{\nu_1} \bigcup_{j=0}^{\nu_2} \bigcup_{k=0}^{\nu_3} \{\ \{(x, y, z), (x', y', z')\} \mid x, x' \in \kappa_1 \ \&\ y, y' \in \kappa_2 \ \&\ z, z' \in \kappa_3\ \}\right)$$

See Figure 1 for a simple illustration of the above definition.

Let $\ell = \lceil 1/\varepsilon \rceil$. It is trivial to compute the $\mathit{\Pi}_{\ell}^{\alpha,\beta,\gamma}[G]$'s for all $0 \leq \alpha, \beta, \gamma < \ell$ in $O(\ell^3|V(G)|)$ time. For each $\mathit{\Pi}_{\ell}^{\alpha,\beta,\gamma}[G]$, OPT$(\mathit{\Pi}_{\ell}^{\alpha,\beta,\gamma}[G], K)$ can be calculated in $O(K2^{\ell^3}|V(G)|)$ time since:



G

(0,0,0)-shifted 2-partition of G

(1,0,0)-shifted 2-partition of G

Fig. 1. Illustration of Definition 2 for a G embeddable in the 2D lattice (*i.e.*, $n_3 = 2$).

- For each $i \in [0, \nu_1]$, $j \in [0, \nu_2]$ and $k \in [\nu_3]$, the subgraph $G_{i,j,k,\alpha,\beta,\gamma}$ of $\Pi_\ell^{\alpha,\beta,\gamma}[G]$ induced by the set of vertices $V(G_{i,j,k,\alpha,\beta,\gamma}) = V(G) \cap \{x, y, z \mid x \in \kappa_1 \ \& \ y \in \kappa_2 \ \& \ z \in \kappa_3\}$ is not connected by any edge of $\Pi_\ell^{\alpha,\beta,\gamma}[G]$ to any remaining vertex of $\Pi_\ell^{\alpha,\beta,\gamma}[G]$. Thus, we can compute $\mathrm{OPT}(G_{i,j,k,\alpha,\beta,\gamma}, \mu)$ for all $1 \le \mu \le K$ by exhaustive enumeration in $O(K2^{\ell^3})$ time. Since there are at most $|V(G)|$ $G_{i,j,k,\alpha,\beta,\gamma}$'s that are not empty, the total time for this step is $O(K2^{\ell^3}|V(G)|)$.
- We now use the dynamic programming algorithm for the General Knapsack (GK) problem. For each $i \in [0, \nu_1]$, $j \in [0, \nu_2]$ and $k \in [0, \nu_3]$, we have a set of K objects $\mathcal{A}_{i,j,k} = \{a^1_{i,j,k}, a^2_{i,j,k}, \ldots, a^K_{i,j,k}\}$ with $s(a^\mu_{i,j,k}) = \mu$ and $v(a^\mu_{i,j,k})$ $=\mathrm{OPT}(G_{i,j,k,\alpha,\beta,\gamma}, \mu)$ for $\mu \in [1, K]$, and moreover we set $\mathbf{b} = K$. We can solve this instance of the GK problem to determine in $O(K|V(G)|)$ time a subset of indices $\{(i_1, j_1, k_1), (i_2, j_2, k_2), \ldots, (i_t, j_t, k_t)\}$ such that $\sum_{p=1}^{t} |V(G_{i_p,j_p,k_p,\alpha,\beta,\gamma})| \le K$ and $\sum_{p=1}^{t} |E(G_{i_p,j_p,k_p,\alpha,\beta,\gamma})|$ is maximized. Obviously,
  $\mathrm{OPT}(\Pi_\ell^{\alpha,\beta,\gamma}[G], K) = \sum_{p=1}^{t} |E(G_{i_p,j_p,k_p,\alpha,\beta,\gamma})|$.

Our algorithm then outputs $\max_{\alpha,\beta,\gamma} \mathrm{OPT}(\Pi_\ell^{\alpha,\beta,\gamma}[G], K)$ as the approximate solution. The total time taken by the algorithm is therefore $O(K2^{\ell^3}\ell^3|V(G)|) = O(K|V(G)|)$ since $\varepsilon > 0$ is a constant. We now show that
$\max_{\alpha,\beta,\gamma} \mathrm{OPT}(\Pi_\ell^{\alpha,\beta,\gamma}[G], K) \ge \left(1 - \frac{1}{\ell}\right)\mathrm{OPT}(G, K) \ge (1 - \varepsilon)\mathrm{OPT}(G, K)$. For each $0 \le \alpha, \beta, \gamma < \ell$, let $E_{-}(\alpha, \beta, \gamma) = E_{-} - E(\Pi_\ell^{\alpha,\beta,\gamma}[G])$, $E_{|}(\alpha, \beta, \gamma) = E_{|} - E(\Pi_\ell^{\alpha,\beta,\gamma}[G])$ and $E_{/}(\alpha, \beta, \gamma) = E_{/} - E(\Pi_\ell^{\alpha,\beta,\gamma}[G])$. Now we observe the following:

- The sets $E_{-}(\alpha, \beta, \gamma)$, $E_{|}(\alpha, \beta, \gamma)$ and $E_{/}(\alpha, \beta, \gamma)$ are mutually disjoint.
- For any $e \in E_{-}$ (respectively, $e \in E_{|}$, $e \in E_{/}$), $|\{E_{-}(\alpha, \beta, \gamma) \mid e \in E_{-}(\alpha, \beta, \gamma)\ \}| \le \ell^2$ (respectively, $|\{E_{|}(\alpha, \beta, \gamma) \mid e \in E_{|}(\alpha, \beta, \gamma)\ \}| \le \ell^2$, $|\{E_{/}(\alpha, \beta, \gamma) \mid e \in E_{/}(\alpha, \beta, \gamma)\ \}| \le \ell^2$). We prove the case for $e \in E_{-}$ only; the other cases are similar. Suppose that $e \in E_{-}(\alpha, \beta, \gamma)$ for some α, β and γ. Then, $e \notin E_{-}(\alpha', \beta', \gamma')$ if $\alpha' \neq \alpha$.

– Thus, $\sum_{\alpha=0}^{\ell-1}\sum_{\beta=0}^{\ell-1}\sum_{\gamma=0}^{\ell-1}\mathrm{OPT}(\mathit{\Pi}_\ell^{\alpha,\beta,\gamma}[G],K)$ is at least

$$\begin{aligned}&\ell^3\mathrm{OPT}(G,K)-\textstyle\sum_{\alpha=0}^{\ell-1}\sum_{\beta=0}^{\ell-1}\sum_{\gamma=0}^{\ell-1}(E_{-}(\alpha,\beta,\gamma)+E_{|}(\alpha,\beta,\gamma)+E_{/}(\alpha,\beta,\gamma)\\ \geq{}&\ell^3\mathrm{OPT}(G,K)-\ell^2(|E_{-}|+|E_{|}|+|E_{/}|)\geq\ell^3\mathrm{OPT}(G,K)-\ell^2\mathrm{OPT}(G,K)\end{aligned}$$

Hence, $\max_{\alpha,\beta,\gamma}\mathrm{OPT}(\mathit{\Pi}_\ell^{\alpha,\beta,\gamma}[G],K)\geq\mathrm{OPT}(G,K)-\frac{1}{\ell}\mathrm{OPT}(G,K)$.

Remark 1. The PTAS can be generalized in an obvious manner when the given graph is embeddable in a d-dimensional lattice for $d > 3$; however the running time grows exponentially with d. We do not describe the generalization here since it has no applications to the PSD problem.

Acknowledgements Berman was supported by NSF grant CCR-O208821, DasGupta was supported in part by NSF grants CCR-0296041, CCR-0206795 and CCR-0208749, Mubayi was supported by NSF grant DMS-9970325, Sloan and Turán were supported by NSF grant CCR-0100336 and Zhang was supported by NSF grant CCR-0208749.

References

1. Y. Asahiro, K. Iwama,H. Tamaki and T. Tokuyama. *Greedily Finding a Dense Subgraph*, Journal of Algorithms 34,203-221,2000.
2. Y. Asahiro, R. Hassin and K. Iwama. *Complexity of finding dense subgraphs*, Discrete Applied Mathematics 121, 15-26,2002.
3. J. Atkins and W. E. Hart. *On the intractability of protein folding with a finite alphabet of amino acids*, Algorithmica, 25(2-3):279–294, 1999.
4. J. Banavar, M. Cieplak, A. Maritan, G. Nadig, F. Seno, and S. Vishveshwara. *Structure-based design of model proteins*, Proteins: Structure, Function, and Genetics, 31:10–20, 1998.
5. B. Berger and T. Leighton. *Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete*, Journal of Computational Biology, 5(1):27–40, 1998.
6. P. Berman, B. DasGupta and S. Muthukrishnan. *Approximation Algorithms For* `MAX-MIN` *Tiling*, Journal of Algorithms, 47 (2), 122-134, July 2003.
7. P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis. *On the complexity of protein folding*, Journal of Computational Biology, 423–466, 1998.
8. J. M. Deutsch and T. Kurosky. *New algorithm for protein design*, Physical Review Letters, 76:323–326, 1996.
9. K. A. Dill, S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas, and H. S. Chan. *Principles of protein folding — A perspective from simple exact models*, Protein Science, 4:561–602, 1995.
10. K. E. Drexler. *Molecular engineering: An approach to the development of general capabilities for molecular manipulation*, Proceedings of the National Academy of Sciences of the U.S.A., 78:5275–5278, 1981.
11. U. Feige and M. Seltser. *On the densest k-subgraph problems.* Technical Report # CS97-16, Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Israel (available online at `http://citeseer.nj.nec.com/feige97densest.html`).

12. W. E. Hart. *On the computational complexity of sequence design problems*, Proceedings of the 1st Annual International Conference on Computational Molecular Biology, 128–136, 1997.
13. W. E. Hart and S. Istrail. *Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal*, Journal of Computational Biology, 3(1):53–96, 1996.
14. W. E. Hart and S. Istrail. *Invariant patterns in crystal lattices: Implications for protein folding algorithms (extended abstract)*, Lecture Notes in Computer Science 1075: Proceedings of the 7th Annual Symposium on Combinatorial Pattern Matching, 288–303, 1996.
15. W. E. Hart and S. Istrail. *Lattice and off-lattice side chain models of protein folding: Linear time structure prediction better than 86% of optimal*, Journal of Computational Biology, 4(3):241–260, 1997.
16. V. Heun. *Approximate protein folding in the HP side chain model on extended cubic lattices*, Lecture Notes in Computer Science 1643: Proceedings of the 7th Annual European Symposium on Algorithms, 212–223, 1999.
17. D. Hochbaum. *Approximation Algorithms for NP-hard problems*, PWS Publishing Company, 1997.
18. D. S. Hochbaum and W. Mass. *Approximation schemes for covering and packing problems in image processing and VLSI*, Journal of ACM, 32(1):130–136, 1985.
19. J. Kleinberg. *Efficient Algorithms for Protein Sequence Design and the Analysis of Certain Evolutionary Fitness Landscapes.*, Proceedings of the 3rd Annual International Conference on Computational Molecular Biology, 226-237, 1999.
20. K. F. Lau and K. A. Dill. *A lattice statistical mechanics model of the conformational and sequence spaces of proteins*, Macromolecules, 22:3986–3997, 1989.
21. G. Mauri, G. Pavesi, and A. Piccolboni. *Approximation algorithms for protein folding prediction*, Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms, 945–946, 1999.
22. K. M. Merz and S. M. L. Grand, editors. *The Protein Folding Problem and Tertiary Structure Prediction*, Birkhauser, Boston, MA, 1994.
23. J. Ponder and F. M. Richards. *Tertiary templates for proteins*, Journal of Molecular Biology, 193:63–89, 1987.
24. S. J. Sun, R. Brem, H. S. Chan, and K. A. Dill. *Designing amino acid sequences to fold with good hydrophobic cores*, Protein Engineering, 8(12):1205–1213, Dec. 1995.
25. E. I. Shakhnovich and A. M. Gutin. *Engineering of stable and fast-folding sequences of model proteins*, Proc. Natl.Acad.Sci., 90:7195-7199, 1993.
26. T. F. Smith, L. L. Conte, J. Bienkowska, B. Rogers, C. Gaitatzes, and R. H. Lathrop. *The threading approach to the inverse protein folding problem*, Proceedings of the 1st Annual International Conference on Computational Molecular Biology, 287–292, 1997.
27. K. Yue and K. A. Dill. *Inverse protein folding problem: Designing polymer sequences*, Proceedings of the National Academy of Sciences of the U.S.A., 89:4163–4167, 1992.