The Inverse Protein Folding Problem on 2D and 3D Lattices*

Piotr Berman†
Department of Computer Science & Engineering
Pennsylvania State University
University Park, PA 16802
Email: berman@cse.psu.edu

Bhaskar DasGupta‡
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607-7053
Email: dasgupta@cs.uic.edu

Dhruv Mubayi§
Department of Mathematics, Statistics & Computer Science
University of Illinois at Chicago
Chicago, IL 60607-7045
Email: mubayi@math.uic.edu

Robert Sloan
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607-7053
Email: sloan@cs.uic.edu

György Turán
Department of Mathematics, Statistics & Computer Science
University of Illinois at Chicago
Chicago, IL 60607-7045
Email: gyt@uic.edu

Yi Zhang¶
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607-7053
Email: yzhang3@cs.uic.edu

November 4, 2005

Abstract

In this paper we investigate the *inverse protein folding* (IPF) problem under the Canonical model on 3D and 2D lattices [13, 26]. In this problem, we are given a contact graph $G = (V, E)$ of a protein sequence that is embeddable in a 3D (respectively, 2D) lattice and an integer $1 \leq K \leq |V|$. The goal is to find an *induced* subgraph of G of at most K vertices with the *maximum* number of edges. In this paper, we prove the following results:

*A preliminary version of this paper without many proofs appeared in 15th Annual Combinatorial Pattern Matching Symposium, LNCS 3109, C. S. Sahinalp, S. Muthukrishnan and U. Dogrusoz (editors), pp. 244-253, July 2004.

†Supported by NSF grant CCR-O208821.

‡Supported by NSF grants CCR-0296041, CCR-0206795 and CCR-0208749.

§Supported by NSF grant DMS-9970325.

¶Supported in part by NSF grant CCR-0208749.

- An earlier proof of NP-completeness of the IPF problem on 3D lattices [13] is based on the NP-completeness of the IPF problem on the 2D lattices. However, the reduction was not correct and we show that the IPF problem for 2D lattices can be solved in $O(K|V|)$ time. But, we show that the IPF problem on 3D lattices is indeed NP-complete by a providing a different reduction from a different NP-complete problem.
- We design a polynomial-time approximation scheme for the IPF problem on 3D lattices using the shifted slice-and-dice approach in [6, 18, 19], thereby improving the previous best polynomial-time approximation algorithm which had a performance ratio of $\frac{1}{2}$ [13].

1 Introduction and Problem Definitions

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, 13–17, 22, 23, 27, 28]. 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 protein landscape is established by a model that relates protein sequences S to protein structures P. Typically this is given by a real-valued function $\Phi : S \times P \rightarrow \mathbb{R}$ that models the "fit" of a sequence $s \in S$ to a structure $p \in P$ with respect 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.* structure q fits 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 *inverse protein folding* (IPF) problem: given a target 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 [28] 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 IPF problem has been investigated in a number of studies [4, 8, 10, 13, 20, 24–26, 28]. The computational complexity of IPF 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 IPF problem is by defining a *heuristic sequence design* (HSD) problem where a simplified pair-wise interaction function is used to compute the landscape function Φ. The implicit assumption is that a sequence that satisfies the HSD problem also solves IPF. Several quantitative models have been proposed for the HSD problem in the literature [8, 25, 26]. This paper is concerned with the Canonical model of Shahknovich and Gutin [26]. This model is specified by **(1)** a geometric representation of a target protein structure with n amino acid residues, **(2)** a *binary folding code* in which the amino acids are classified as *hydrophobic* (H) or *polar* (P) [9, 21], and **(3)** a *fitness function* Φ defined in terms of the target structure that favors sequences

with a dense hydrophobic core and penalizes 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, a H-H residue contact[1] is given a value of -1 and all other contacts are given the value of 0. To prevent the solution from being an 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. This gives rise to the following *special case* of the *densest subgraph problem* on K vertices:

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.* $\{u,v\} \in E(n,3)$*) 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 ($\mathbf{IPFC_2}$/$\mathbf{IPFC_3}$ Problems) *The IPF problem for the Canonical model on a 2D (resp. 3D) Euclidean lattice, denoted by IPFC$_2$ (resp. IPFC$_3$), is an instance of the DS problem when the input graph G is the contact graph realized by a 2D (resp. 3D) sequence.*

Once a solution to the IPCF$_2$/IPCF$_3$ problem is obtained, we can simply label the vertices in V' by H and the rest of the vertices by P to obtain a solution to the original protein sequence design problem.

References [1, 2] consider the DS problem for general graphs. Hart [13] considers both IPFC$_2$ and IPFC$_3$ problems, provides approximation algorithm for IPFC$_3$ with an approximation ratio of $\frac{1}{2}$ and an *almost* optimal algorithm for IPFC$_2$. The following property of the contact graph of a 2D/3D sequence is easy to observe [13]:

> 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 Basic Definitions and Notations

We will use the following notations, definitions and conventions consistently throughout the rest of the paper. G is the given input graph in our problems. $V(H)$ (resp. $E(H)$) is the vertex set (resp. edge set) of any graph H. 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. $H \backslash S$ 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,

[1] A contact in a conformation $p_1, p_2, \ldots, p_n$ correspond to monomers i and j where $|j-i| > 1$ and the Eucildean distance between p_i and p_j is 1.

y and z are the 1st, 2nd and 3rd 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. $\mathrm{OPT}(G, K)$ denote the number of edges in an optimal solution to the IPFC_2 or IPFC_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.

For subsequent usage, we state the *General Knapsack* (GK) problem and its known pseudo-polynomial-time solution. An input to this problem consists of a positive integer $\mathbf{b}$ and a collection of sets of objects $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_m$ where each $a \in \cup_{i=0}^{m} \mathcal{A}_i$ has a *size* (positive integer) $s(a)$ and a *value* (positive integer) $v(a)$. The goal is to select a subset of objects $\mathcal{A}' \subseteq \cup_{i=0}^{m} \mathcal{A}_i$ such that $\sum_{a \in \mathcal{A}'} s(a) \leq \mathbf{b}$, $|\mathcal{A}' \cap \mathcal{A}_i| \leq 1$ for each $i \in [0, m]$ and the total value of selected objects $\sum_{a \in \mathcal{A}'} v(a)$ is *maximized*. A special case of the GK problem is the *subset-sum* problem wherein we wish to find any subset $\mathcal{A}'$ such that $\sum_{a \in \mathcal{A}'} s(a) = \mathbf{b}$. The GK problem or the subset-sum problem is NP-complete; however a $O(|\cup_{i=0}^{m} \mathcal{A}_i|\mathbf{b})$ *pseudo-polynomial* time algorithm via dynamic programming to solve the problem can be designed [12]; in fact this algorithm provides a solution for *every* instance $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_m, \mathbf{b}'$ of the problem for all $0 \leq \mathbf{b}' \leq \mathbf{b}$.

1.2 Our Results

Our results are as follows:

(I) There exists an $O(K|V(G)|)$ time algorithm that solves the IPFC_2 problem (see Section 2).

(II) The IPFC_3 decision problem is NP-complete (see Section 3.1).

(III) For the IPFC_3 problem we can design a PTAS, *i.e.* for any given constant $\varepsilon > 0$, we can design a $O(K|V(G)|)$ time algorithms with a performance ratio of $1-\varepsilon$ (see Section 3.2).

1.3 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, 18, 19].

1.4 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 [20, 25]. 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^{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.

2 The IPFC$_2$ Problem

In [13] Hart provided a proof of NP-completeness of IPFC$_2$. 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 IPFC$_2$ can indeed be solved in polynomial time.

Lemma 2 *There exists an $O(K|V(G)|)$ time algorithm that solves the IPFC$_2$ problem.*

Proof. Our lemma can be proved by using additional arguments in Proposition 2 of [13][2]. Since G has at most two vertices of degree 3 and remaining vertices of degree at most 2, G has at most one connected component with two cycles and remaining connected components with at most one cycle. Thus, $\mathrm{OPT}(G,K) \leq \frac{1}{2}(2(K-2)+6) = K+1$. Using depth-first-search (DFS), one can find the connected components of G in $O(|V(G)|+|E(G)|) = O(|V(G)|)$ time. Classify a connected component of G as of the i^{th} type if it contains exactly i cycles for $0 \leq i \leq 2$. These components have the following properties:

• G has at most one component of the 2^{nd} type. Moreover, such a component C consists of two cycles C_1 and C_2 that either share one simple path of one or more edges or are connected by one simple path of one or more edges. Define a partial cover[3] of C to be either an empty set or consists of a connected subgraph of C that contains at least one of C_1 or C_2 but not both; a partial cover of C with x vertices has therefore exactly x edges.

• All but two of the connected components of G of the 1^{st} type are simple cycles; define a partial cover of a simple cycle to be the entire simple cycle. The at most two remaining connected components which are not simple cycles consist of a simple cycle C with a simple path attached to one vertex of C; define a partial cover of such a component to be either an empty set or a connected subgraph of it that contains the cycle C.

The above observations lead us to the following cases:

Case 1: $K \geq n_2(G) + n_1(G)$. Then, an optimal solution contains all vertices in connected components of G of the 1^{st} and the 2^{nd} type. Moreover, if $K > n_2(G) + n_1(G)$, we create a sorted list T of the connected components of the 0^{th} type in decreasing order of their number of vertices, greedily pick all vertices in connected components from T from the beginning until our total number of vertices y exceed K. If $y > K$ we greedily remove $y - K$ vertices from the last connected component selected from T such that the remaining vertices from this component form a *connected* subgraph of the component. Suppose that we selected from t 0^{th} type connected components. Then, our solution has $(n_2(G)+n_1(G)+1)+(K-(n_2(G)+n_1(G))-t) = K+1-t$ edges. On the other hand, $\mathrm{OPT}(G,K) \leq K+1-t$ since it must use vertices from at least t 0^{th} type components.

Case 2: $n_2(G) \leq K < n_2(G) + n_1(G)$. We select all the $n_2(G)$ vertices in the components of the 2^{nd} type. If $K > n_2(G)$, then it suffices to select an additional $K - n_2(G)$ vertices from the

[2]Hart [13] showed that an almost optimal bound of 1+OPT(G,K) can be achieved in $O(|V(G)|)$ time.

[3]Partial covers should not be confused with the usual vertex covers for graphs despite similarity of names.

components of the 1st type. Let C_1 and C_2 be those at most two connected components of G of the 1st type that are not simple cycles (one or both of C_1 and C_2 may be empty), and let $C_3, C_4, \ldots, C_p$ be the remaining connected components of the 1st type ($\sum_{i=1}^{p} |V(C_i)| = n_1(G)$). Let

$$L = \{\ell \mid \ell = \alpha_\ell + \beta_\ell,\ C_1 \text{ and } C_2 \text{ has partial covers with } \alpha_\ell \text{ and } \beta_\ell \text{ vertices, respectively}\}$$

We use the dynamic programming algorithm for the subset-sum problem to determine, for all $\ell \in L$, if there is a subset of indices $\{i_1, i_2, \ldots, i_t\} \subseteq [3, p]$ such that $\sum_{j=1}^{t} |V(C_{i_j})| = K - n_2(G) - \ell$. Since $K - n_2(G) - \ell \in [0, n_1(G)]$ for any $\ell \in L$, the total time taken is $O(p(K - n_1(G))) = O(K|V(G)|)$. There are now two subcases:

Case 2.1: there is such a subset of indices corresponding to some $\ell \in L$. Then, our solution includes the additional $K - n_2(G) - \ell$ vertices of $C_{i_1}, \ldots, C_{i_t}$, a partial cover of C_1 of α_ℓ vertices and a partial cover of C_2 of β_ℓ vertices. This is an optimal solution since it has $K + 1$ edges.

Case 2.2: there is no such subset of indices. Our solution has to include at least two vertices of degree 1 (corresponding to the two end vertices of a path resulting from at least one simple cycle could not be covered completely) and we need to minimize the number of such vertices. We create a sorted list T of $C_1, C_2, C_3, C_4, \ldots, C_p$ in decreasing order of their number of vertices, greedily pick all vertices in each connected subgraph from T from the beginning until our total number of vertices y exceed K, and then greedily remove $K - y$ vertices from the last connected component selected from T such that the remaining vertices from this component form a *connected* subgraph of the component. This is an optimal solution since we select exactly two vertices of degree 1.

Case 3: $K < n_2(G)$**.** This case implies that G has one connected component C of the 2nd type, all connected components of G of the 1st type are simple cycles and $K - 1 \leq \mathrm{OPT}(G, K) \leq K$. Let $C_1, C_2, \ldots, C_p$ be the connected components of G of the 1st type. We use the dynamic programming algorithm for the subset-sum problem to determine in $O(pK) = O(K|V(G)|)$ time, for all $0 \leq \alpha < n_2(G)$ such that C has a partial cover of α vertices, if there is a subset of indices $\{i_1, i_2, \ldots, i_t\} \subseteq [1, p]$ such that $\sum_{j=1}^{t} |V(C_{i_j})| = K - \alpha$. Now, again, there are two subcases.

Case 3.1: there is such a subset of indices corresponding to some α. Then, our solution includes the $K - \alpha$ vertices of $C_{i_1}, \ldots, C_{i_t}$ and a partial cover of C of α vertices. This is an optimal solution since it has K edges.

Case 2.2: there is no such subset of indices. This implies that $\mathrm{OPT}(G, K) = K - 1$. We select any connected subgraph of C containing K vertices. ❑

3 The IPFC$_3$ Problem

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

3.1 NP-completeness Result for IPFC$_3$

Theorem 3 *The IPFC$_3$ problem is NP-complete.*

Proof. It is trivial to see that IPFC$_3$ 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[4]. 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:

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 $\mathcal{S}$.

Below we provide these two steps.

Step 1 (embedding H in the 3D lattice):

We show that H is a subgraph of a 3D lattice $(V(\text{poly}(n), 3), E(\text{poly}(n), 3))$ when poly(n) denotes a polynomial in n. It is trivial to see that a connected component with no vertex of degree greater than 2 that is not an odd cycle is a subgraph of the 3D lattice, so we concentrate on the graph $H' = H \setminus (\cup_{i=1}^s Q_i)$. We use the following notations in the rest of the proof:

- $(x_1, y_1, z_1) \rightarrow (x_2, y_2, z_2)$ denotes a path of $|x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2|$ edges from vertex (x_1, y_1, z_1) to vertex (x_2, y_2, z_2) in the 3D lattice in which all edges that connect vertices that differ in their i^{th} coordinates precede all edges that connect vertices that differ in their j^{th} coordinates if $i < j$.
- For $1 \leq i < j \leq n$, define δ_{ij} by $2\delta_{ij} = kt_5 - (j - i)(t_4 + t_3) - 2jt_2 - 2it_1$. Note that δ_{ij} is a positive even integer since n is a sufficiently large power of 2.

We embed H' in the 3D lattice as follows (see Figure 1):

[4]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.

- Cycle vertex v_j^i of C^i (for $j \in [1, 2nt_4]$) are mapped to

$$f(v_j^i) = \begin{cases} (it_3, j, 0) & \text{if } j \in [1, nt_4] \\ (it_3 + 1, j - nt_4, 0) & \text{if } j \in [nt_4 + 1, 2nt_4] \end{cases}$$

Edges of C^i (for each $i \in [1, n]$) are mapped to the cycle consisting of the set of edges

$$\left(\bigcup_{j \in [1, 2nt_4] \setminus \{nt_4\}} \{f(v_j^i), f(v_{j+1}^i)\} \right) \cup \{f(v_1^i), f(v_{nt_4+1}^i)\} \cup \{f(v_{nt_4}^i), f(v_{2nt_4}^i)\}$$

- The path vertices and edges in each path P^{ij} are mapped to the 3D lattice as:

$$\begin{aligned} (x_1^{ij}, y_1^{ij}, 0) \to (x_2^{ij}, y_1^{ij}, 0) \to (x_2^{ij}, y_1^{ij}, \delta_{ij} + 1) \to (x_2^{ij}, y_2^{ij}, \delta_{ij} + 1) \\ \downarrow \\ (x_4^{ij}, y_2^{ij}, 0) \leftarrow (x_3^{ij}, y_2^{ij}, 0) \leftarrow (x_3^{ij}, y_2^{ij}, \delta_{ij} + 1) \end{aligned}$$

where $x_1^{ij} = it_3 + 1$, $y_1^{ij} = jt_4$, $x_2^{ij} = i(t_3 + t_2) + jt_1$, $y_2^{ij} = it_4$. $x_3^{ij} = j(t_3 + t_2) + it_1$, and $x_4^{ij} = jt_3 + 1$. The number of edges $|P^{ij}|$ of the path is precisely

$$\begin{aligned} &(it_2 + jt_1 - 1) + \delta_{ij} + 1 + (j - i)t_4 + ((j - i)(t_3 + t_2) + (i - j)t_1) \\ &+\delta_{ij} + 1 + (jt_2 + it_1 - 1) = kt_5 + 1 \end{aligned}$$

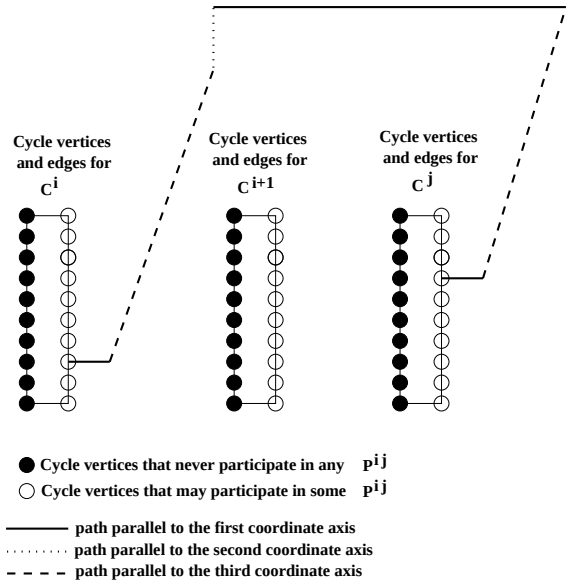

Figure 1: Pictorial illustrations of embeddings of cycle C^i and path P^{ij}.

We also need to show that no two distinct vertices of H' are mapped to the same vertex in the 3D lattice. For this purpose, the following proposition and its corollary would be very useful.

Proposition 4 *Consider two numbers $x = \alpha_0 + \sum_{i=1}^{5} \alpha_i t_i$ and $y = \beta_0 + \sum_{i=1}^{5} \beta_i t_i$, where $\alpha_i, \beta_i \in [0, 4n^2]$ for $i \in [0, 5]$. Then, $x = y$ if and only if $\alpha_i = \beta_i$ for all i.*

Proof. Each α_i and β_i can be represented as a $2 + 2\log_2 n$ bit binary number (possibly with leading zeros) and multiplying α_i or β_i by t_i adds $\log_2 t_i \gg 2 + 2\log_2 n$ trailing zeros to the binary representation of α_i or β_i. ❑

Corollary 5 *For two distinct edges* $\{v_i, v_j\}, \{v_{i'}, v_{j'}\} \in E(G)$, $\delta_{ij} \neq \delta_{i'j'}$.

Proof. Notice that $2j, j-i \in [0, 2n] \subset [0, 4n^2]$. Thus, $\delta_{ij} \neq \delta_{i'j'}$ because either $j \neq j'$ or, if $j = j'$ then $i \neq i'$ and thus $j - i \neq j - i'$. ❑

It is obvious that no two cycle vertices are mapped to the same vertex in the 3D lattice and it is also easy to verify no path vertex is identical to any cycle vertex (since $n(t_1 + t_2) < t_3$). We show below that mappings of no two distinct paths P^{ij} and $P^{i'j'}$ share any path vertices:

- Any path vertex u on the subpath $(x_1^{ij}, y_1^{ij}, 0) \rightarrow (x_2^{ij}, y_1^{ij}, 0) \rightarrow (x_2^{ij}, y_1^{ij}, \delta_{ij}+1)$ and $(x_3^{ij}, y_2^{ij}, \delta_{ij}+1) \rightarrow (x_3^{ij}, y_2^{ij}, 0) \rightarrow (x_4^{ij}, y_2^{ij}, 0)$ is not the same as any path vertex v in $P^{i'j'}$ because either $i \neq i'$ or $j \neq j'$ and thus we can use Proposition 4 (if necessary) to show that either the 1^{st} coordinate or the 2^{nd} coordinate of u and v are distinct.
- Any path vertex u on the subpath $(x_2^{ij}, y_1^{ij}, \delta_{ij} + 1) \rightarrow (x_2^{ij}, y_2^{ij}, \delta_{ij} + 1) \rightarrow (x_3^{ij}, y_2^{ij}, \delta_{ij} + 1)$ is not the same as any path vertex v in $P^{i'j'}$ because $\delta_{i,j} > 0$ and $\delta_{i,j} \neq \delta_{i',j'}$ by Corollary 5.

Step 2 (realizing H as a contact graph):

We can design a sequence $\mathcal{S}$ in three stages as follows:

Stage 1: For each $i \in [1, n]$, we design a sequence whose contact graph consists of the "cycle" edges $\Big(\bigcup_{j\in[1,nt_4-1]}\{(it_3, j, 0), (it_3, j+1, 0)\}\Big) \cup \Big(\bigcup_{j\in[nt_4+1,2nt_4-1]}\{(it_3+1, j-nt_4, 0), (it_3+1, j-nt_4+1, 0)\}\Big) \cup \{(it_3, 1, 0), (it_3+1, 1, 0)\} \cup \{(it_3, nt_4.0), (it_3, 2nt_4, 0)\}$, the first and the last path edge of each path P^{ij} for all $\{v_i, v_j\} \in E(G)$ with $i < j$ and some additional connected components that are part of $Q_1, \ldots, Q_s$.

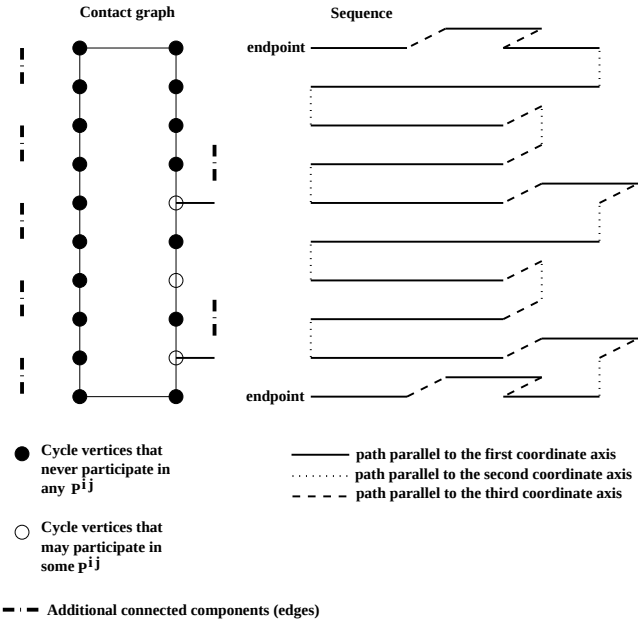

Figure 2: Embedding the cycle edges and the first and last edges of each path.

Let J_i be the set of indices such that the edge $\{v_i, v_j\}$ is in $E(G)$. Note that, by our construction, if $i < j$ then the path P^{ij} begins at $(it_3 + 1, jt_4, 0)$ whereas if $i > j$ then the path P^{ij} ends at $(it_3 + 1, jt_4, 0)$, jt_4 (for all j) is a positive even integer since n is a sufficiently large power of 2 and any two indices in J_i differ by at least t_4.

For each $j \in [1, nt_4]$, let $\mathcal{S}^{ij}$ be the sequence given by[5]

$$\begin{array}{rl}
(it_3-1,1,0) \to (it_3,1,0) \to (it_3,1,1) \to (it_3+1,1,1) & \text{if } j = 1 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \downarrow & \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (it_3+2,1,0) \leftarrow (it_3+1,1,0) & \\
(it_3+2,nt_4,0) \to (it_3+1,nt_4,0) \to (it_3+1,nt_4,1) & \text{if } j = nt_4 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \downarrow & \\
\qquad\qquad\qquad\qquad\qquad\qquad (it_3-1,nt_4,0) \leftarrow (it_3,nt_4,0) \leftarrow (it_3,nt_4,1) & \\
\qquad\qquad\qquad\qquad\qquad\qquad (it_3-1,j,0) \to (it_3+2,j,0) & \text{if } j \equiv 1 \pmod 2 \text{ and } j-1 \notin J_i \\
\qquad\qquad\qquad\qquad (it_3-1,j,0) \to (it_3+1,j,0) \to (it_3+1,j,1) & \text{if } j \equiv 1 \pmod 2 \text{ and } j-1 \in J_i \\
\qquad\qquad\qquad\qquad\qquad\qquad (it_3+2,j,0) \to (it_3-1,j,0) & \text{if } j \equiv 0 \pmod 2 \text{ and } j, j-2 \notin J_i \\
\qquad\qquad\qquad\qquad (it_3+1,j,1) \to (it_3+1,j,0) \to (it_3-1,j,0) & \text{if } j \equiv 0 \pmod 2 \text{ and } j-2 \notin J_i \\
\qquad\qquad\qquad\qquad (it_3+2,j,0) \to (it_3+2,j,1) \to (it_3+1,j,1) & \text{if } j \in J_i \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \downarrow & \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (it_3-1,j,0) \leftarrow (it_3+1,j,0) &
\end{array}$$

Then, our desired sequence $\mathcal{S}_i$ is given by $\mathcal{S}^{i,1} \to \mathcal{S}^{i,2} \to \cdots \to \mathcal{S}^{i,nt_4}$. We refer to $(it_3-1,1,0)$ and $(it_3-1,nt_4,0)$ as the as the two *endpoints* of this $\mathcal{S}_i$. See Figure 2 for a pictorial illustration.

Stage 2: For each $\{v_i, v_j\} \in E(G)$ with $i < j$, we design a sequence $\mathcal{T}^{ij}$, whose contact graph realizes the path edges of P^{ij} *excluding the first and the last edges*, namely the edges $(x_1^{ij}+1, y_1^{ij}, 0) \to (x_2^{ij}, y_1^{ij}, 0) \to (x_2^{ij}, y_1^{ij}, \delta_{ij}+1) \to (x_2^{ij}, y_2^{ij}, \delta_{ij}+1) \to (x_3^{ij}, y_2^{ij}, \delta_{ij}+1) \to (x_3^{ij}, y_2^{ij}, 0) \to (x_4^{ij}+1, y_2^{ij}, 0)$ and some additional connected components that are part of $Q_1, \ldots, Q_s$.

A path in which adjacent vertices differ in exactly the same i^{th} coordinate, such as $(x,y,z) \to (x+1,y,z) \to (x+2,y,z) \to \cdots$, can be realized (with additional connected components of vertices of degree no greater than 2) as a contact graph of a sequence that also varies one of the remaining two coordinates, *e.g.* see Figure 3. Similarly, a path that can be partitioned into two such subpaths in two different coordinates, such as $(x,y,z) \to (x+100,y,z) \to (x+100,y+50,z)$, can also be realized (with additional connected components of vertices of degree no greater than 2) by the concatenation of two such above sequences with a corner gadget, *e.g.* see Figure 3. Using this approach, it is possible to design in a straightforward but *extremely tedious* manner the sequence $\mathcal{T}^{ij}$. We refer to $(x_1^{ij}+2, y_1^{ij}, 0)$ and $(x_4^{ij}+2, y_2^{ij}, 0)$ as the two *endpoints* of $\mathcal{T}^{i,j}$.

Stage 3: Now we connect the endpoints of the subsequences $\mathcal{S}_i$'s and $\mathcal{T}^{i,j}$'s without introducing any crossings such that a complete sequence $\mathcal{S}$ is obtained. Let $(\alpha_1,\alpha_2),(\alpha_3,\alpha_4),\ldots,(\alpha_{2r-1},\alpha_{2r})$ be the endpoints of the r subsequences for the $\mathcal{S}_i$'s and $\mathcal{T}^{i,j}$'s. We connect α_{2i} and α_{2i+1} (for $i \in [1,r)$) as $\alpha_{2i} = (x,y,0) \to (x,y,-it_6) \to (x',y,-it_6) \to (x',y',-it_6) \to (x',y',0) = \alpha_{2i+1}$. The additional connected components created are added to $Q_1, Q_2, \ldots, Q_s$. ❑

Corollary 6 *The 3DS problem is NP-complete even if G is a subgraph of the 3D lattice.*

[5]We make use of our assumption that the vertex v_n has zero degree and thus the vertex $(it_3+1, nt_4.0)$ cannot participate in any path P^{ij}, and, by construction, the vertex $(it_3+1, 1.0)$ does not participate in any P^{ij} either.

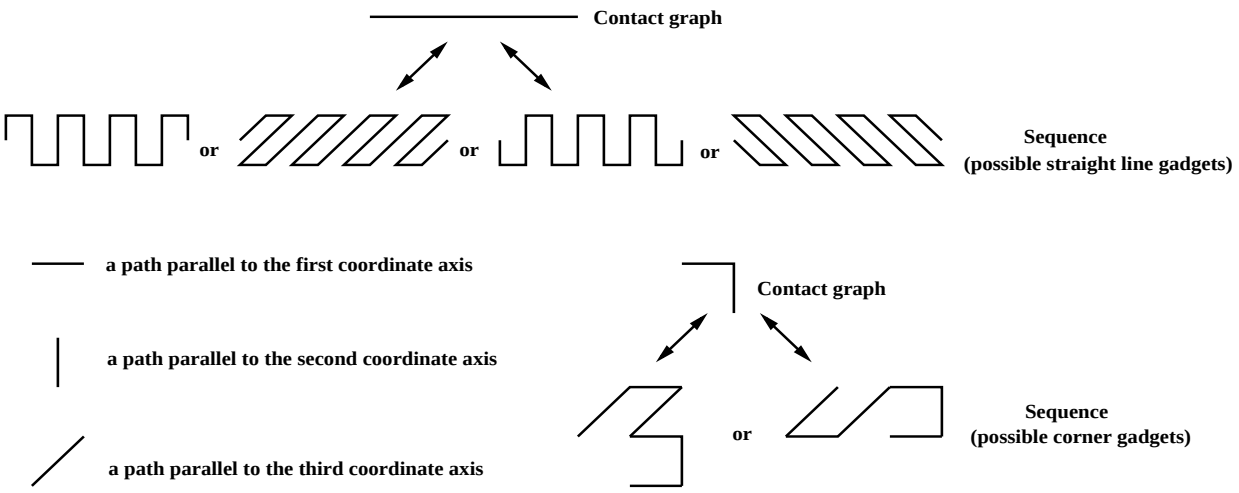

Figure 3: Some components of a sequence to embed the path P^{ij} excluding its first and last edges.

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 $E_{\text{–}}$, $E_{|}$ and $E_{/}$ be the set of horizontal, vertical and lateral edges in an optimal solution.

Theorem 7 *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 IPFC$_3$ problem with at least $(1-\varepsilon)OPT(G, K)$ edges.*

Proof. We use the shifted slice-and-dice technique of [6, 18, 19]. 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 8 *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 $\Pi_\ell^{\alpha,\beta,\gamma}[G]$ is the subgraph of G in which $V(\Pi_\ell^{\alpha,\beta,\gamma}[G]) = V(G)$ and $E(\Pi_\ell^{\alpha,\beta,\gamma}[G])$ 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 4 for a simple illustration of the above definition.

Let $\ell = \lceil 1/\varepsilon \rceil$. It is trivial to compute the $\Pi_\ell^{\alpha,\beta,\gamma}[G]$'s for all $0 \leq \alpha, \beta, \gamma < \ell$ in $O(\ell^3 |V(G)|)$ time. For each $\Pi_\ell^{\alpha,\beta,\gamma}[G]$, $\text{OPT}(\Pi_\ell^{\alpha,\beta,\gamma}[G], K)$ can be calculated in $O(K 2^{\ell^3} |V(G)|)$ time since:

- 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

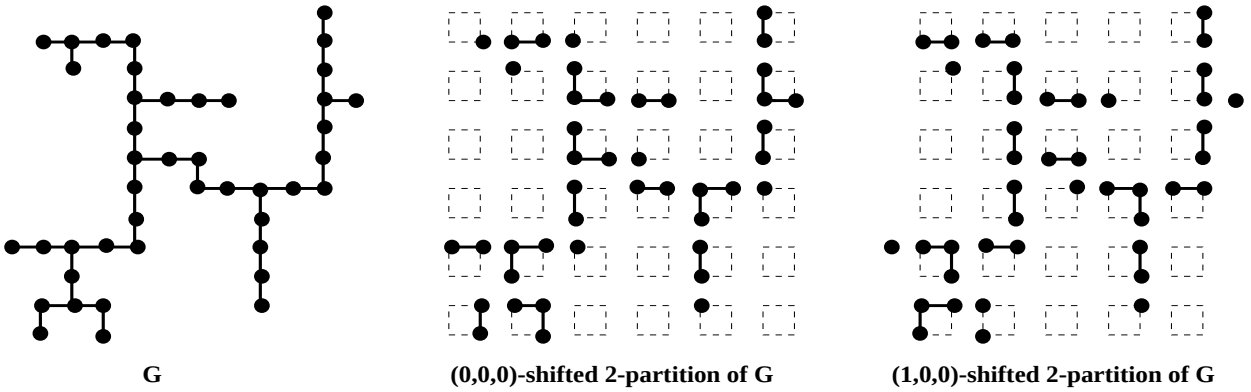

Figure 4: Illustration of Definition 8 for a G embeddable in the 2D lattice (*i.e.*, $n_3 = 2$).

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 $1 \leq \mu \leq K$ 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 $\sum_{p=1}^t |V(G_{i_p,j_p,k_p,\alpha,\beta,\gamma})| \leq K$ 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. Figure 5 illustrates the approach used in the above algorithm.

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 $\max_{\alpha,\beta,\gamma}\mathrm{OPT}(\Pi_\ell^{\alpha,\beta,\gamma}[G], K) \geq \left(1 - \frac{1}{\ell}\right)\mathrm{OPT}(G, K) \geq (1 - \varepsilon)\mathrm{OPT}(G, K)$. For each $0 \leq \alpha, \beta, \gamma < \ell$, let $E_{\text{–}}(\alpha, \beta, \gamma) = E_{\text{–}} - E(\Pi_\ell^{\alpha,\beta,\gamma}[G])$, $E_{|}(\alpha, \beta, \gamma) = E_{|} - E(\Pi_\ell^{\alpha,\beta,\gamma}[G])$ and $E_{/}(\alpha, \beta, \gamma) = E_{/} - E(\Pi_\ell^{\alpha,\beta,\gamma}[G])$. Now we observe the following:

- The sets $E_{\text{–}}(\alpha, \beta, \gamma)$, $E_{|}(\alpha, \beta, \gamma)$ and $E_{/}(\alpha, \beta, \gamma)$ are mutually disjoint.
- For any $e \in E_{\text{–}}$ (respectively, $e \in E_{|}$, $e \in E_{/}$), $|\{E_{\text{–}}(\alpha, \beta, \gamma) \mid e \in E_{\text{–}}(\alpha, \beta, \gamma)\}| \leq \ell^2$ (respectively, $|\{E_{|}(\alpha, \beta, \gamma) \mid e \in E_{|}(\alpha, \beta, \gamma)\}| \leq \ell^2$, $|\{E_{/}(\alpha, \beta, \gamma) \mid e \in E_{/}(\alpha, \beta, \gamma)\}| \leq \ell^2$). We prove the case for $e \in E_{\text{–}}$ only; the other cases are similar. Suppose that $e \in E_{\text{–}}(\alpha, \beta, \gamma)$ for some α, β and γ. Then, $e \notin E_{\text{–}}(\alpha', \beta', \gamma')$ if $\alpha' \neq \alpha$.
- Thus, $\sum_{\alpha=0}^{\ell-1}\sum_{\beta=0}^{\ell-1}\sum_{\gamma=0}^{\ell-1}\mathrm{OPT}(\Pi_\ell^{\alpha,\beta,\gamma}[G], K)$ is at least

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

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

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 IPF problem.*

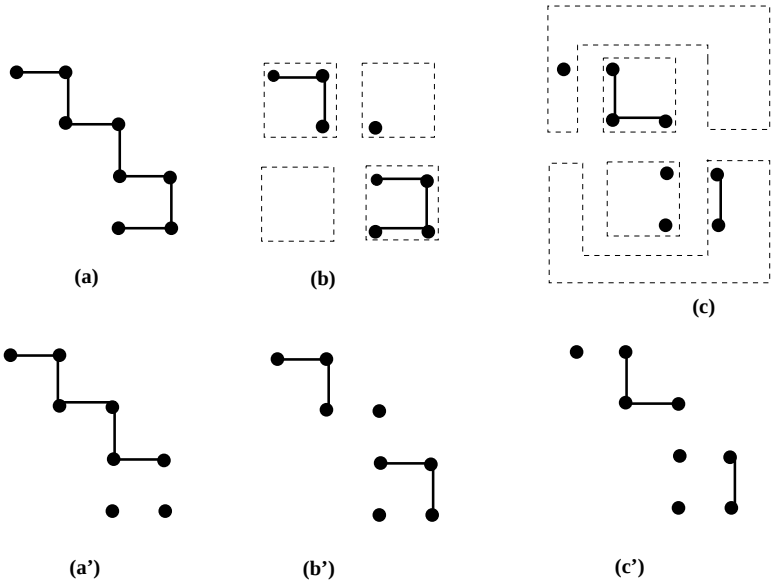

Figure 5: Illustration of the technique used in our algorithm for a G embeddable in the 2D lattice with $K = 6$ and $\ell = 2$. **(a)** and **(a')** The given graph and an optimal solution with $\mathrm{OPT}(G, 6) = 5$. **(b)** $\Pi_2^{0,0,0}[G]$. **(b')** $\mathrm{OPT}(\Pi_2^{0,0,0}[G], K) = 4$. **(c)** $\Pi_2^{1,0,0}[G]$. **(c')** $\mathrm{OPT}(\Pi_2^{1,0,0}[G], K) = 3$.

Remark 2 *The running time of the PTAS may be slightly improved with a more careful implementation of the shifted slice-and-dice technique.*

Remark 3 *It suffices to set $\ell = 3$ to improve upon the $\frac{1}{2}$-approximation algorithm of Hart* [13].

Acknowledgements

We would like to thank both the reviewers for their comments which improved the presentaion of the paper.

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] M. R. Garey and D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] D. Hochbaum. *Approximation Algorithms for NP-hard problems*, PWS Publishing Company, 1997.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] K. M. Merz and S. M. L. Grand, editors. *The Protein Folding Problem and Tertiary Structure Prediction*, Birkhauser, Boston, MA, 1994.

[24] J. Ponder and F. M. Richards. *Tertiary templates for proteins*, Journal of Molecular Biology, 193:63–89, 1987.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

APPENDIX

In this appendix, we provide more details about the sequence $\mathcal{T}^{ij}$ in the proof of Theorem 3, for each $\{v_i, v_j\} \in E(G)$ with $i < j$, whose contact graph realizes the path edges of P^{ij} excluding the first and the last edges, namely the edges $(x_1^{ij}+1, y_1^{ij}, 0) \to (x_2^{ij}, y_1^{ij}, 0) \to (x_2^{ij}, y_1^{ij}, \delta_{ij}+1) \to (x_2^{ij}, y_2^{ij}, \delta_{ij}+1) \to (x_3^{ij}, y_2^{ij}, \delta_{ij}+1) \to (x_3^{ij}, y_2^{ij}, 0) \to (x_4^{ij}+1, y_2^{ij}, 0)$ along with some additional connected components that are part of $Q_1, \ldots, Q_s$. We will use the following notations:

• A *direction* is an element of $\{X^+, X^-, Y^+, Y^-, Z^+, Z^-\}$. Directions d^+ and d^- are opposite of each other for any $d \in \{X, Y, Z\}$.

• By k steps in the direction of X^+ (resp. X^-) from a vertex (x, y, z) we mean the path $(x, y, z) \to (x+k, y, z)$ (resp. $(x, y, z) \to (x-k, y, z)$); k steps in the directions of Y^+, Y^-, Z^+ and Z^- from (x, y, z) are analogously defined on the second and third coordinates of (x, y, z).

• $(x_1, y_1, z_1)^{d_1} \rightrightarrows^{d_2} (x_2, y_2, z_2)^{d_1}$ denotes a sequence that starts at (x_1, y_1, z_1), first goes one step in the d_1 direction, then goes one step in the d_2 direction, two steps in the opposite of d_1 direction, one step in d_2 direction, two steps in the d_1 direction, one step in the d_2 direction, two steps in the opposite of d_1 direction, …, until it reaches (x_2, y_2, z_2).

• The notation $(x_1, y_1, z_1)^{d_1} \Longrightarrow (x_2, y_2, z_2)^{d_2}$ is defined as follows. For convenience we assume $x_1 \le x_2$, $y_1 \le y_2$ and $z_1 \le z_2$; the definition is similar for other cases. $(x_1, y_1, z_1)^{d_1} \Longrightarrow (x_2, y_2, z_2)^{d_2}$ defined a sequence S whose contact graph H is as defined below and moreover any vertex (α, β, γ) of S satisfies the relationship as described below for each corresponding H:

- if $|x_1 - x_2| = 2$, $|y_1 - y_2| = 2$ and $|z_1 - z_2| = 0$, H is $(x_1, y_1, z_1) \to (x_2, y_1, z_1) \to (x_2, y_2, z_2)$, and $x_1 \le \alpha \le x_2$, $y_1 \le \beta \le y_2$, $z_1 - 1 \le \gamma \le z_1 + 1$;
- if $|x_1 - x_2| = 2$, $|y_1 - y_2| = 0$ and $|z_1 - z_2| = 2$, H is $(x_1, y_1, z_1) \to (x_2, y_1, z_1) \to (x_2, y_2, z_2)$, and $x_1 \le \alpha \le x_2$, $y_1 - 1 \le \beta \le y_1 + 1$, $z_1 \le \gamma \le z_2$;
- if $|x_1 - x_2| = 0$, $|y_1 - y_2| = 2$ and $|z_1 - z_2| = 2$, H is $(x_1, y_1, z_1) \to (x_1, y_2, z_1) \to (x_2, y_2, z_2)$, and $x_1 - 1 \le \alpha \le x_2 + 1$, $y_1 \le \beta \le y_2$, $z_1 \le \gamma \le z_2$;
- otherwise, the notation is undefined.

Satisfying the above constraints, the sequence $(x_1, y_1, z_1)^{d_1} \Longrightarrow (x_2, y_2, z_2)^{d_2}$ can be stated as:

- for $x_2 = x_1 + 2 = x$, $y_1 = y_2 - 2 = y$ and $z_2 = z_1 = z$:
  - if $d_1 \in \{Y^+, Z^-\}$ and $d_2 \in \{Z^+, X^+, X^-\}$ then $(x-2, y, z)^{d_1} \Longrightarrow (x, y+2, z)^{d_2}$ is

$$
\begin{array}{ccccccccc}
 & & (x-2,y,z) & \to & (x-2,y+1,z) & \to & (x-1,y+1,z) & \to & (x-1,y,z) \\
 & & & & & & & & \downarrow \\
(x,y+1,z-1) & \leftarrow & (x,y+1,z+1) & \leftarrow & (x,y,z+1) & \leftarrow & (x,y,z-1) & \leftarrow & (x-1,y,z-1) \\
\downarrow & & & & & & & & \\
(x,y+2,z-1) & \to & (x,y+2,z) & & & & & &
\end{array}
$$

  - if $d_1 \in \{Y^-, Z^+\}$ and $d_2 \in \{Z^+, X^+, X^-\}$ then $(x-2, y, z)^{d_1} \Longrightarrow (x, y+2, z)^{d_2}$ is

$$
\begin{array}{ccccccccc}
(x-2,y,z) & \to & (x-2,y,z+1) & \to & (x-1,y,z+1) & \to & (x-1,y,z-1) & \to & (x,y,z-1) \\
 & & & & & & & & \downarrow \\
(x,y+2,z) & \leftarrow & (x,y+2,z-1) & \leftarrow & (x,y+1,z-1) & \leftarrow & (x,y+1,z+1) & \leftarrow & (x,y,z+1)
\end{array}
$$

– if $d_1 = Z^+$ and $d_2 = Z^-$, the sequence is similar to the sequence for $d_1 = Z^-$ and $d_2 = Z^+$;

– if $d_1 \in \{X^+, X^-, Y^+, Y^-\}$ and $d_2 = Z^-$, the sequence is symmetric to the sequence with the same d_1 and $d_2 = Z^+$.

- For other (x_1, y_1, z_1) and (x_2, y_2, z_2) satisfying that two of $|x_1 - x_2|$, $|y_1 - y_2|$ and $|z_1 - z_2|$ are 2 and the third one is 0, we can easily make similar sequences as above and whose contact graph is H.

With all these new notations, we can write sequence $\mathcal{T}^{ij}$ whose contact graph realizes the path edges of P^{ij} excluding the first and the last edges:

• if $y_1 > y_2$ and δ_{ij} is an odd number, $\mathcal{T}^{ij}$ is $(x_1^{ij}+2, y_1^{ij}, 0)^{Y^+} \rightrightarrows^{X^+} (x_2^{ij}-2, y_1^{ij}, 0)^{Y^+} \Longrightarrow (x_2^{ij}, y_1^{ij}, 2)^{X^+} \rightrightarrows^{Z^+} (x_2^{ij}, y_1^{ij}, \delta_{ij}-1)^{X^+} \Longrightarrow (x_2^{ij}, y_1^{ij}-2, \delta_{ij}+1)^{X^+} \rightrightarrows^{Y^-} (x_2^{ij}, y_2^{ij}+2, \delta_{ij}+1)^{X^+} \Longrightarrow (x_2^{ij}+2, y_2^{ij}, \delta_{ij}+1)^{Y^-} \rightrightarrows^{X^+} (x_3^{ij}-2, y_2^{ij}, \delta_{ij}+1)^{Y^-} \Longrightarrow (x_3^{ij}, y_2^{ij}, \delta_{ij}-1)^{X^+} \rightrightarrows^{Z^-} (x_3^{ij}, y_2^{ij}, 2)^{X^+} \Longrightarrow (x_3^{ij}-2, y_2^{ij}, 0)^{Z^-} \rightrightarrows^{X^-} (x_4^{ij}+2, y_2^{ij}, 0)^{Z^-}$.

• If $y_1 > y_2$ and δ_{ij} is an even number, $\mathcal{T}^{ij}$ is $(x_1^{ij}+2, y_1^{ij}, 0)^{Y^+} \rightrightarrows^{X^+} (x_2^{ij}-2, y_1^{ij}, 0)^{Y^+} \Longrightarrow (x_2^{ij}, y_1^{ij}, 2)^{X^+} \rightrightarrows^{Z^+} (x_2^{ij}, y_1^{ij}, \delta_{ij}-2)^{X+} \rightarrow (x_2^{ij}+1, y_1^{ij}, \delta_{ij}-2) \rightarrow (x_2^{ij}+1, y_1^{ij}, \delta_{ij}-1) \rightarrow (x_2^{ij}, y_1^{ij}, \delta_{ij}-1)^{X^-} \Longrightarrow (x_2^{ij}, y_1^{ij}-2, \delta_{ij}+1)^{X^+} \rightrightarrows^{Y^-} (x_2^{ij}, y_2^{ij}+2, \delta_{ij}+1)^{X^+} \Longrightarrow (x_2^{ij}+2, y_2^{ij}, \delta_{ij}+1)^{Y^-} \rightrightarrows^{X^+} (x_3^{ij}-2, y_2^{ij}, \delta_{ij}+1)^{Y^-} \Longrightarrow (x_3^{ij}, y_2^{ij}, \delta_{ij}-1)^{X^+} \rightrightarrows^{Z^-} (x_3^{ij}, y_2^{ij}, 3)^{X^+} \rightarrow (x_3^{ij}+1, y_2^{ij}, 3) \rightarrow (x_3^{ij}+1, y_2^{ij}, 2) \rightarrow (x_3^{ij}, y_2^{ij}, 2)^{X^-} \Longrightarrow (x_3^{ij}-2, y_2^{ij}, 0)^{Z^-} \rightrightarrows^{X^-} (x_4^{ij}+2, y_2^{ij}, 0)^{Z^-}$.

• If $y_1 < y_2$ and δ_{ij} is an odd number, $\mathcal{T}^{ij}$ is $(x_1^{ij}+2, y_1^{ij}, 0)^{Y^+} \rightrightarrows^{X^+} (x_2^{ij}-2, y_1^{ij}, 0)^{Y^+} \Longrightarrow (x_2^{ij}, y_1^{ij}, 2)^{X^+} \rightrightarrows^{Z^+} (x_2^{ij}, y_1^{ij}, \delta_{ij}-1)^{X^+} \Longrightarrow (x_2^{ij}, y_1^{ij}+2, \delta_{ij}+1)^{X^+} \rightrightarrows^{Y^+} (x_2^{ij}, y_2^{ij}-2, \delta_{ij}+1)^{X^+} \Longrightarrow (x_2^{ij}+2, y_2^{ij}, \delta_{ij}+1)^{Y^+} \rightrightarrows^{X^+} (x_3^{ij}-2, y_2^{ij}, \delta_{ij}+1)^{Y^+} \Longrightarrow (x_3^{ij}, y_2^{ij}, \delta_{ij}-1)^{X^+} \rightrightarrows^{Z^-} (x_3^{ij}, y_2^{ij}, 2)^{X^+} \Longrightarrow (x_3^{ij}-2, y_2^{ij}, 0)^{Z^-} \rightrightarrows^{X^-} (x_4^{ij}+2, y_2^{ij}, 0)^{Z^-}$.

• If $y_1 < y_2$ and δ_{ij} is an even number, $\mathcal{T}^{ij}$ is $(x_1^{ij}+2, y_1^{ij}, 0)^{Y^+} \rightrightarrows^{X^+} (x_2^{ij}-2, y_1^{ij}, 0)^{Y^+} \Longrightarrow (x_2^{ij}, y_1^{ij}, 2)^{X^+} \rightrightarrows^{Z^+} (x_2^{ij}, y_1^{ij}, \delta^{ij}-2)^{X^+} \rightarrow (x_2^{ij}+1, y_1^{ij}, \delta^{ij}-2) \rightarrow (x_2^{ij}+1, y_1^{ij}, \delta^{ij}-1) \rightarrow (x_2^{ij}, y_1^{ij}, \delta_{ij}-1)^{X^-} \Longrightarrow (x_2^{ij}, y_1^{ij}+2, \delta_{ij}+1)^{X^+} \rightrightarrows^{Y^+} (x_2^{ij}, y_2^{ij}-2, \delta_{ij}+1)^{X^+} \Longrightarrow (x_2^{ij}+2, y_2^{ij}, \delta_{ij}+1)^{Y^+} \rightrightarrows^{X^+} (x_3^{ij}-2, y_2^{ij}, \delta_{ij}+1)^{Y^+} \Longrightarrow (x_3^{ij}, y_2^{ij}, \delta_{ij}-1)^{X^+} \rightrightarrows^{Z^-} (x_3^{ij}, y_2^{ij}, 3) \rightarrow (x_3^{ij}+1, y_2^{ij}, 3) \rightarrow (x_3^{ij}+1, y_2^{ij}, 2) \rightarrow (x_3^{ij}, y_2^{ij}, 2)^{X^-} \Longrightarrow (x_3^{ij}-2, y_2^{ij}, 0)^{Z^-} \rightrightarrows^{X^-} (x_4^{ij}+2, y_2^{ij}, 0)^{Z^-}$.