Journal of Universal Computer Science, vol. 6, no. 6 (2000), 560-579
submitted: 2/4/00, accepted: 15/5/00, appeared: 28/6/00 © Springer Pub. Co.

Invariant Patterns in Crystal Lattices: Implications for Protein Folding Algorithms[1]

William E. Hart
(Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185-1110
Email: wehart@sandia.gov)

Sorin Istrail
(Celera Genomics
45 West Gude Drive
Rockville, MD 20850
Email: Sorin.Istrail@celera.com)

Abstract: Crystal lattices are infinite periodic graphs that occur naturally in a variety of geometries and which are of fundamental importance in polymer science. Discrete models of protein folding use crystal lattices to define the space of protein conformations. Because various crystal lattices provide discretizations of the same physical phenomenon, it is reasonable to expect that there will exist "invariants" across lattices related to fundamental properties of the protein folding process. This paper considers whether performance-guaranteed approximability is such an invariant for HP lattice models. We define a master approximation algorithm that has provable performance guarantees provided that a specific sublattice exists within a given lattice. We describe a broad class of crystal lattices that are approximable, which further suggests that approximability is a general property of HP lattice models.

Key Words: Protein folding, lattice models, HP model, approximation algorithm.
Category: J.3,F.2

1 Introduction

Crystal lattice models are vehicles for reasoning about the protein folding phenomenon through analogy. Crystal lattices are infinite periodic graphs that are generated by translations of a "unit cell" that fill a two or three-dimensional space. In polymer science many important results have been obtained through the use of lattice models [9, 17]. In the context of protein folding, lattices provide a natural discretization of the space of protein conformations. The sequence of amino acids that defines a protein can be viewed as a path labeled with amino acids on vertices. A conformation of a protein is a self-avoiding embedding of this path into a lattice, where each vertex of the path is mapped to a vertex of the lattice and edges of the path are mapped to edges of the lattice. With every conformation we can associate an energy value using rules defined by the model, which take into account the neighborhood relationship of the amino acids.

[1] This paper is part of the January special issue of JUCS on *Automata, Logic, and Computability dedicated to Professor Sergiu Rudeanu Festschrift*, edited by C.S. Calude and G. Ştefănescu.

In this paper we consider algorithms for protein structure prediction for crystal lattice models. Lattices models of protein folding have provided valuable insight into the general complexity of protein structure prediction problems. For example, protein structure prediction has been shown to be NP-hard for a variety of lattice models [3, 4, 6, 13]. This lends credibility to the general assumption that protein structure prediction is an intractible problem. These results are complemented by analyses of protein folding algorithms that prove worst-case performance guarantees for a variety of lattice models [1, 7, 12, 15]. These results show that near-optimal protein structures can be quickly constructed, and they can be generalized to simple off-lattice protein models [15].

Of particular interest here is the design of algorithms that can be applied to a variety of lattice models. Results that transcend particular lattice frameworks are of significant interest because they can say something about the general biological problem with a higher degree of confidence. In fact, it is reasonable to expect that there will exist algorithmic invariants across lattices that fundamentally relate to the protein folding problem, because lattice models provide discretizations of the same physical phenomenon.

We have previously addressed the issue of algorithmic invariance in our hardness results for lattice models [13]. This analysis considers a simple empirical potential model that uses a distance-related energy with an unbounded number of amino acid types [22]. Our results extend the NP-hardness argument of Unger and Moult [22] to all three-dimensional lattices that have a single, infinite connected component. This result provides stronger evidence for the intractibility of protein folding problems because of its independence from a specific lattice formulation.

This paper considers whether performance guaranteed approximation algorithms can be applied to a wide range of lattice models. We consider approximation algorithms for the hydrophobic-hydrophilic model. This model categorizes amino acids as hydrophobic (nonpolar) or hydrophilic (polar), and the energy of a conformation is equal to the number of hydrophobic-hydrophobic contacts. We describe two "master" approximation algorithms that can be applied to lattices that contain a general sublattice that we call a latticoid. Latticoids impose a structure in which a skeleton of hydrophobic contacts can be constructed, thereby leading to folding algorithms whose performance can be analyzed. In the particular case of the square two-dimensional lattice, the latticoid describes the structure used in the approximation algorithms described by Hart and Istrail [12].

We prove that our master approximation algorithms have performance guarantees for a class of lattices that includes most of the lattices commonly used in simple exact protein folding models, e.g. two- and three-dimensional square lattice [9, 11, 19], the diamond (carbon) lattice [20], the face-centered-cubic lattice [5] and the 210 lattice used by Skolnick [21]. Furthermore, this class encompasses a large number of other lattices studied in crystallography. These results extend and consolodate our previous results in Hart and Istrail [14].

2 Lattice Models for Protein Folding

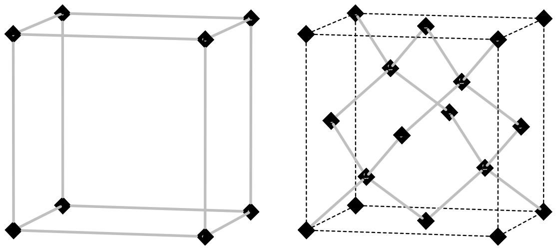
Lattice models for protein folding can be distinguished by at least five properties:

1. An alphabet of types of amino acids that the model considers;

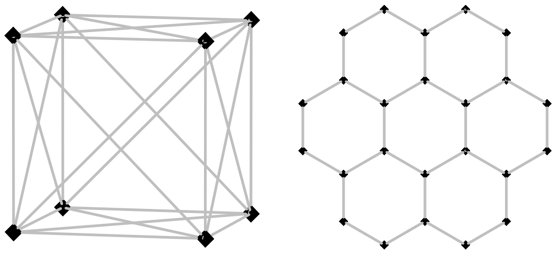
2. The set of protein instances represented as sequences from this alphabet;
3. An energy formula specifying how the conformational energy is computed;
4. Parameters for the energy formula;
5. A crystal lattice that provides a discretization of the conformation space.

For example, the hydrophobic-hydrophilic (HP) model [8] can be described as follows. The alphabet used in an HP model is $A = \{0, 1\}$, and the set of protein instances is the set of binary sequences $\sigma = \{0, 1\}^+$. Each sequence $s \in \sigma$ is the (hypothesized) hydrophobic-hydrophilic pattern of a protein sequence, where 1 represents a hydrophobic amino acid, and 0 represents a hydrophilic amino acid. We will refer to s as a protein instance. Contact energies are used in this model, so the energy formula is an energy matrix, $\mathcal{E}$. The energy matrix is indexed by the alphabet symbols, $\mathcal{E} = (e(a, b))_{a,b \in A}$. For HP models, $e(a, b) = -1$ if $a = b = 1$, and $e(a, b) = 0$ otherwise. Conformations for the HP model have been commonly studied for the a square or cubic lattices.

We consider protein folding models on a large class of crystal lattices, including the square lattice. Crystal lattices are infinite periodic graphs that are generated by translations of a "unit cell" that fill a two- or three-dimensional space (e.g., see Ashcroft and Mermin [2]). Examples of unit cells for crystal lattices are shown in Figure 1.



(a) (b)



(c) (d)

Figure 1: Examples of crystal lattices: (a) cubic, (b) diamond, (c) cubic with planar diagonals, and (d) hexagonal.

We can characterize crystal lattices in graph-theoretic terms as follows. A *unit cell* is a volume of space (in two or three dimensions), that can be translated to fill all of space, such that

1. the volume contains a graph with finitely many points
2. edges that pass through the surface of the volume connect graphs in neighboring unit cells.

From this definition, it follows that connectivity between unit cells is symmetric. Consider three adjacent unit cells generated by translating a unit cell in a single direction, $c_1c_2c_3$. If there is an edge connecting c_1 and c_2, then there must exist an similar edge connecting c_2 and c_3.

Let G be an infinite periodic graph generated by translations of a unit cell. G is *connected* if there exists a path between any two vertices in G. Consider the graph derived from G in which vertices represent unit cells and edges represent a connection between two unit cells. If G is connected then this corresponding graph is connected, which is a property common to physical crystal structures. An *(ideal) crystal lattice*, L, is a connected infinite periodic graph generated by translations of a unit cell. A *sublattice* $\hat{L}$ of L is a subgraph of L that is obtained by removing edges and vertices from L.

One can interpret a protein sequence $s = s_1...s_m$ as an m-vertex node-labeled path, where for $1 \leq i \leq m$, node i is labeled with s_i. The path has $m-1$ edges that are called *bonds.* A *conformation* C of a protein sequence s in a lattice L is a path in the lattice in which the protein sequence is embedded, i.e., the protein vertices are mapped one-to-one to lattice points, and protein bonds are mapped to the corresponding lattice edges. The *energy* of a conformation of the protein sequence s in L is typically computed using distances in the lattice. For example, in the HP model the energy is a function of the number of "contact edges." A contact edge is a lattice edge that is not a protein bond (in the embedding) but has both endpoints labeled. In HP models, contact edges with 1s at their endpoints have weight -1 while all other contact edges have weight 0.

The *native* conformation of a protein is the conformation that has biological function. According to the Thermodynamic Hypothesis the native conformation of a protein is the conformation with the minimum energy among the set of all conformations. Consequently, given a sequence s and a lattice model, the protein folding structure prediction problem is to find a native conformation of s in L. This problem is known to be NP-hard for the square and cubic lattices [4, 6], but performance-guaranteed approximation algorithms have been developed for several common lattices (e.g. square, cubic and face-centered-cubic lattices).

Let $\mathcal{Z}_L(s)$ be the energy of the conformation generated for protein instance s on lattice L by algorithm $\mathcal{Z}_L$, and let $OPT_L(s)$ be the energy of the optimal conformation of s on L. A standard performance guarantee used for approximation algorithms is the asymptotic performance ratio $R^\infty(\mathcal{Z}_L)$ [10]. If $R^\infty(\mathcal{Z}_L) = \tau$, then as $\mathcal{Z}_L$ is applied to larger protein instances, the value of solutions generated by $\mathcal{Z}_L$ approaches a factor of τ of the optimum. Here, "large" protein instances have low conformational energy at their native state, which may be independent of their length. Since $\mathcal{Z}_L(s) \leq 0$ and $OPT_L(s) \leq 0$, both of these ratios are scaled between 0 and 1 such that a ratio closer to 1 indicates better performance.

The following lemma will be used to prove asymptotic performance guarantees for the approximation algorithms that we consider.

Lemma 1. *Let $\mathcal{A}$ be an approximation algorithm such that for a sequence s $\mathcal{A}(s) \leq -Af(s) + B$, for constants $A > 0$ and $B \geq 0$, and for a function f such that $f(s) \geq 0$ for all s. If $OPT_L(s) \geq -Cf(s) - D$, for constants $C > 0$ and $D \geq 0$, then $R^\infty(\mathcal{A}) \geq A/C$.*

Proof. From the definition of $R_\mathcal{A}(s)$ we have

$$R_\mathcal{A}(s) = \frac{\mathcal{A}(s)}{OPT_L(s)} \geq \frac{-Af(s) + B}{-Cf(s) - D}. \tag{1}$$

Let $S_N = \{s \mid OPT(s) \leq N\}$ and $R^N_\mathcal{A} = \inf\{R_\mathcal{A}(s) \mid s \in S_N\}$. For $s \in S_N$, $f(s) \geq -(N + D)/C$. Since $R_\mathcal{A}(s)$ is monotonically increasing for $f(s) \geq 0$, we have

$$R_\mathcal{A}(s) \geq \frac{A(N + D)/C + B}{N + 2D},$$

for $s \in S_N$. Thus

$$R^N_\mathcal{A} \geq \frac{A(N + D)/C + B}{N + 2D},$$

and from the definition of $R^\infty(\mathcal{A})$ [10] we have

$$R^\infty(\mathcal{A}) = \sup\{r \mid R^N_\mathcal{A} \geq r, N \in \mathbf{Z}\} \geq \lim_{N\to\infty} R^N_\mathcal{A} = A/C.$$

3 Master Approximation Algorithms for the HP Model

We now describe two paradigms for designing master approximation algorithms for the HP model that can be applied to a wide range of lattices. HP models abstract the hydrophobic interaction process in protein folding by reducing a protein to a heteropolymer that represents a predetermined pattern of hydrophobicity in the protein. This is one of the most studied lattice models for protein folding, and despite its simplicity, the model is powerful enough to capture a variety of properties of actual proteins [9].

The first master approximation algorithm that we describe captures two aspects of the protein folding algorithms described by Hart and Istrail [12]: (1) the selection of a folding point that balances hydrophobicity and (2) the skeleton of contact edges that forms the hydrophobic core. We call this the *bipartite master approximation algorithm* because it is applicable to crystal lattices that can be described as a bipartite graph. These crystal lattices have the property that two 1's can be endpoints of a contact edge only if there is an even number of elements between them.

The second master approximation algorithm is related to approximation algorithms that have been developed for the triangular lattice [1]. For lattices that contain odd-length cycles, each hydrophobic amino acid can often be placed adjacent to all other hydrophobic amino acids in the chain. We call these lattices nonbipartite to reflect that the lattice does not emplicitly enforce a bipartite labeling the hydrophobics, and to provide a performance guarantee for a nonbipartitie lattice it suffices to generate a chain of contacts that connect a fraction of all hydrophobics in a protein sequence. The *nonbipartite master approximation algorithm* is applicable to nonbipartite lattices to form such a chain.

3.1 Protein Sequence Structure in the HP Model

This section summarizes key definitions concerning the structure of protein instances from Hart and Istrail [12]. Let $s = s_1, \ldots, s_m$ be a protein instance, $s_i \in \{0, 1\}$. Let $l(s)$ equal the length of the sequence s. Let $M_{max}(s)$ equal the length of the longest sequence of zeros in s, and let $M_{min}(s)$ equal the length of the shortest sequence of zeros in s. Finally, let $E(s)$ equal the number of adjacent elements in the sequence, s_j and s_{j+1} for which $s_j = 1$ and $s_{j+1} = 1$.

An instance s can be decomposed into a sequence of *blocks*. A block b_i has the form $b_i = 1$ or $b_i = 1Z_{i_1}1 \ldots Z_{i_k}1$, where the Z_{i_j} are odd-length sequences of 0's and $k \geq 1$. A *block separator* z_i is a sequence of 0's that separates two consecutive blocks, where $l(z_i) \geq 0$ and $l(z_i)$ is even for $i = 1, \ldots, h-1$. Thus s is decomposed into $z_0 b_1 z_1 \ldots b_h z_h$. Since $l(z_i) \geq 0$, this decomposition treats consecutive 1's as a sequence of blocks separated by zero-length block separators. Let $N(b_i)$ equal the number 1's in b_i. Thus the sequence

$$0\,\underbrace{10101}_{b_1}\,\underbrace{1}_{b_2}\,\underbrace{1}_{b_3}\,\underbrace{10101}_{b_4}\,0000\,\underbrace{1010101}_{b_5}$$

gives us $l(z) = (1, 0, 0, 0, 4, 0)$ and $N(b) = (3, 1, 1, 3, 4)$.

It is useful to divide blocks into two categories: x-blocks and y-blocks. For example, let $x_i = b_{2i}$ and let $y_i = b_{2i-1}$. Let B_x and B_y be the number of x-blocks and y-blocks respectively. Further, let $X = X(s) = \sum_{i=1}^{B_x} N(x_i)$ and $Y = Y(s) = \sum_{i=1}^{B_y} N(y_i)$. Let $T_x(s)$ equal the number of endpoints of s that are 1's in x-blocks, and let $T_y(s)$ equal the number of endpoints of s that are 1's in y-blocks. We assume that the division into x- and y-blocks is such that $X \leq Y$ and if $X = Y$ then $T_x(s) \geq T_y(s)$. For example, the sequence

$$0\,\underbrace{10101}_{y_0}\,\underbrace{1}_{x_0}\,\underbrace{1}_{y_1}\,\underbrace{10101}_{x_1}\,0000\,\underbrace{1010101}_{y_2}$$

can be represented as $z_0 y_0 z_1 x_0 z_2 y_1 z_3 x_1 z_4 y_2 z_5$, where $l(z) = (1, 0, 0, 0, 4, 0)$, $N(x) = (1, 3)$, and $N(y) = (3, 1, 4)$.

A *superblock* B_i is comprised of sequences of blocks as follows: $B_i = b_{i_1} z_{i_1} \ldots z_{i_{h-1}} b_{i_h}$. Let $N_x(B_i)$ equal the sum of $N(b_j)$, where b_j are x-blocks in B_i. Let $N_y(B_i)$ equal the sum of $N(b_j)$, where b_j are y-blocks in B_i. Finally, let $N(B_i) = N_x(B_i) + N_y(B_i)$.

Note that two 1's can be endpoints of a contact edge only if there is an even number of elements between them [12]. It follows from our definition of blocks that two 1's within a block cannot be in contact. Further, any pair of 1's take from blocks b_k and b_j may be in contact only when $|k - j|$ is odd. This makes it clear that 1's from an x-block can only be in contact with 1's from an y-block.

3.2 The Bipartite Master Approximation Algorithm

Consider the following definitions.

Definition 1 *Given a path p in a lattice L from a to b, let $d_p(a, b)$ be the length of p. A path p from a to b is* polynomial extensible *if for some $\gamma \in \mathbf{Z}^{>0}$ there exist paths p_k for every $k \in \mathbf{Z}^{>0}$ such that $d_{p_k}(a, b) = d_p(a, b) + \gamma k$ and there* *exists a polynomial time algorithm that given* p *and* k *constructs* p_k. *If* $\gamma = 2$, *then we say that these paths are polynomial evenly extensible. The collection of the paths of a polynomial extensible path* p *is called the* extension *of* p *in* L.

Definition 2 *Given polynomial extensible paths* p *from* a *to* b *and* q *from* c *to* d, *we say that* p *and* q *are* extensibly disjoint *if their extensions are vertex disjoint.*

Definition 3 *A* bipartite latticoid, $\hat{L}$, *of a bipartite lattice* L *is an infinite graph that contains an infinite sequence of contact edges* (a_i, b_i) *with the following properties:*

- *There is a polynomial evenly extensible path* p_i^a *from* a_i *to* a_{i+1} *and polynomial evenly extensible path* p_i^b *from* b_i *to* b_{i+1},
- *There is a constant* $\kappa > 0$ *such that for every* i *and* j, $d_{p_i^a}(a_i, a_{i+1}) = d_{p_j^b}(b_j, b_{j+1}) = 2\kappa$, *and*
- *The set of paths* $\{p_i^a, p_i^b \mid i = 1, \ldots\}$ *are mutually extensibly disjoint.*

The dilation *of the bipartite latticoid is* $\Delta_{\hat{L}} = \kappa$.

Figure 2 illustrates the structure of a bipartite latticoid. Because the paths Ai are evenly extensible, the paths Bi and Ci can be constructed in polynomial time. Furthermore, the vertices in $\{Ai, Bi, Ci\}$ and $\{Aj, Bj, Cj\}$ do not intersect. Figure 3 shows two bipartite latticoids of the two-dimensional square lattice, $\mathbf{L}_0$. The dilation of $\hat{\mathbf{L}}_0^2$ is 2, and the dilation of $\hat{\mathbf{L}}_0^H$ is 3.

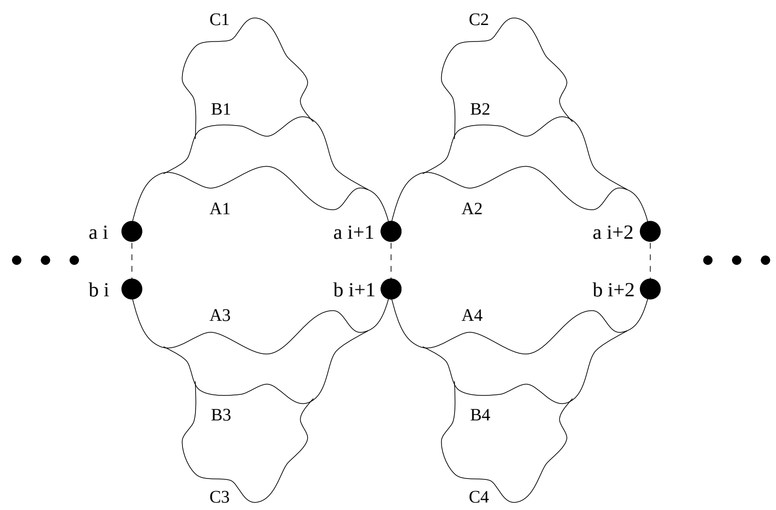
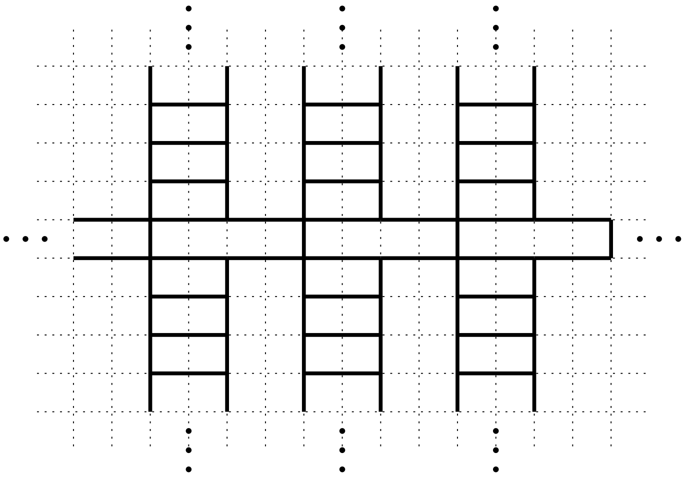
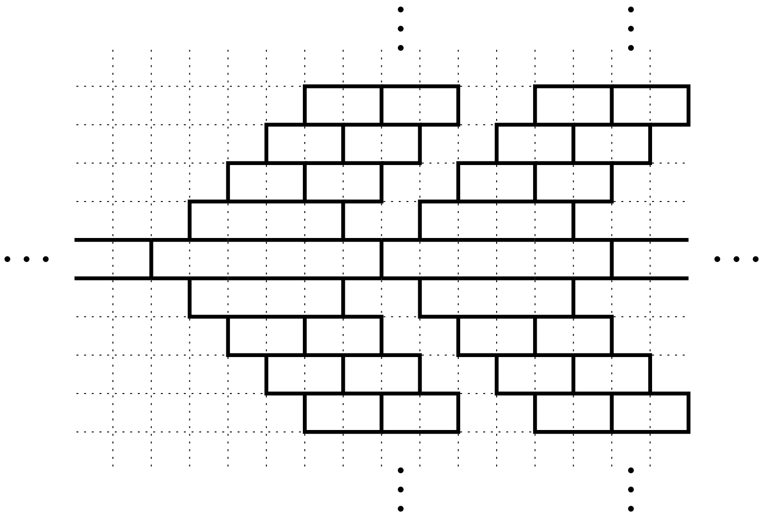


Figure 2: A symbolic illustration of the structure of bipartite latticoids.



(a)



(b)

Figure 3: Two possible bipartite latticoids of $\mathbf{L}_0$: (a) $\hat{\mathbf{L}}_0^2$, and (b) $\hat{\mathbf{L}}_0^H$. Dark lines indicate edges that are used for some protein conformation. Dashed lines indicate the remaining edges in $\mathbf{L}_0$. The contact edges are the vertical edges of the center bolded horizontal row.

The bipartite master approximation algorithm takes a bipartite latticoid $\hat{L}$ and selects a single folding point (turning point) that divides a protein instance into a y-superblock B' and an x-superblock B''. The folding point is selected using "Subroutine 1" from Hart and Istrail [12]. Subroutine 1 selects a folding point that balances the hydrophobicity between the x-blocks and y-blocks on each half of the folding point. The following lemma describes the key property of the folding point that is selected.

Lemma 2 ([12], Lemma 1). *The folding point selected by Subroutine 1 partitions a protein instance s into two superblocks B' and B'' such that either*

$$N_y(B') \geq \lceil (Y+1)/2 \rceil \quad \text{and} \quad N_x(B'') \geq \lceil X/2 \rceil$$
$$\text{or}$$
$$N_y(B') \geq \lceil Y/2 \rceil \quad \text{and} \quad N_x(B'') \geq \lceil (X+1)/2 \rceil .$$

After selecting the folding point, the conformation of the two superblocks is dictated by the bipartite latticoid $\hat{L}$. The bipartite latticoid specifies the placement of the contact edges between the superblocks, as well as the conformation of the loops within each superblock that connect the contact points. These loops follow the path of the polynomially extensible path in the latticoid. The embedded structure of protein sequences in the latticoid generalizes the notion of "normal form" that was used to describe the approximation algorithms in Hart and Istrail [12].

Decomposition into x- and y-blocks requires a single pass through the protein instance, and the selection of the folding point via Subroutine 1 requires linear time. The construction of the final conformation requires polynomial time to create the paths for the loops between contact points. Thus the computation required by Algorithm $\mathcal{A}_{\hat{L}}$ to construct a conformation for a given latticoid is polynomial.

Let $\mathcal{A}_{\hat{L}}(s)$ represent the energy of the final conformation generated by Algorithm $\mathcal{A}_{\hat{L}}$. The performance of Algorithm $\mathcal{A}_{\hat{L}}$ can be bounded as follows.

Lemma 3.

$$\mathcal{A}_{\hat{L}}(s) \leq - \left\lceil \frac{X}{2\Delta_{\hat{L}}} \right\rceil + 1.$$

Proof. Let B' and B'' be the two halves of the protein sequence identified by Subroutine 1, and suppose that B' forms a y-superblock and B'' forms a x-superblock. From Lemma 2 we know that $N_y(B') \geq \lceil X(s)/2 \rceil$ and $N_x(B'') \geq \lceil X(s)/2 \rceil$. On a square or cubic lattice, these two halves of the sequence could be aligned to form at least $\lceil X(s)/2 \rceil$ hydrophobic contacts.

In the dilated latticoid, the minimum distance between consecutive contacts is $2\Delta_L$. Considering B', it follows that there can be $\Delta_L - 1$ y-hydrophobics between y-hydrophobics at contact points (e.g. consider a sequence of the form $(10)^k1$). Thus in the worst case the minimum number of contacts that can be guaranteed is

$$\left\lceil \frac{\lceil X(s)/2 \rceil}{\Delta_{\hat{L}}} \right\rceil + 1 \geq \left\lceil \frac{X(s)}{2\Delta_{\hat{L}}} \right\rceil + 1.$$

Here, we add one to this term to account for the fact that the folding point may be between consecutive 1's in the sequence.

Let $\delta(L)$ be the maximum degree of all vertices in L. Proposition 1 presents the asymptotic performance ratio for Algorithm $\mathcal{A}_{\hat{L}}$ where $\hat{L}$ is a latticoid of L.

Proposition 1 *Let $\hat{L}$ be a latticoid of L. Then $R^\infty(\mathcal{A}_{\hat{L}}) \geq 1/(2\Delta_{\hat{L}}(\delta(L)-2))$.*

Proof. Since L is a crystal lattice generated by a unit cell, $\delta(L)$ is finite. It follows from the fact that L is bipartite that $OPT_L(s) \geq -(\delta(L)-2)X(s)-2$. The bound on $R^\infty(\mathcal{A}_{\hat{L}})$ follows from Lemma 1 and Lemma 3.

To illustrate the application of the bipartite master approximation algorithm, consider its application to the diamond lattice, which has previously been used in lattice models for protein folding (e.g. see [20]). The latticoid $\hat{\mathbf{L}}_0^2$ can be embedded in the diamond lattice as follows. Consider the labeled unit cell for a diamond lattice in Figure 4a. Observe that the cycles (A, F, B, D, G, C, A) and (C, G, D, A, H, B, C) can be embedded into the latticoid $\hat{\mathbf{L}}_0^2$. Figure 4b illustrates this embedding, along with neighbors of the members of these two cycles. To show that all of $\hat{L}_0^2$ can be embedded, we need to extend the sublattice both vertically and horizontally. We can do this by exploiting the relationships between vertices in Figure 4b. The path (D, A, F, B, C) can be extended to a cycle (D, A, F, B, C, I, D) by observing that between every C and D vertex is an I vertex. The path (H, B, D, I) can be extended to a cycle (H, B, D, I, C, A, H) by observing that every I vertex is adjacent to a C vertex and every H vertex is adjacent to an A vertex. Similarly, the path (I, D, A, F) can be extended to the cycle (I, D, A, F, B, C, I). Figure 4c shows the expanded embedding.

To extend the sublattice vertically and horizontally, it suffices to shift the expanded embedding to extend paths to cycles using the cycles that exist in the expanded embedding. It follows that the latticoid $\hat{\mathbf{L}}_0^2$ can be embedded into the diamond lattice, since it is a sublattice of the embedded sublattice. Note that the unit cells used by this embedding comprise one slice through the three-dimensional lattice.

Figure 5 demonstrates this embedding for a particular conformation. Grey and black solid lines between vertices in each unit cell indicate the edges of the diamond lattice that are used to embed a square lattice for which one dimension is dilated to length two. Edges not used for this embedding are omitted. The solid lines illustrate a conformation of a protein on this lattice that the bipartite master approximation algorithm would generate. Now $\delta(L) = 4$ for the diamond lattice L, so it follows from Proposition 1 that $R^\infty(\mathcal{A}_L) = 1/8$.

3.3 The Nonbipartite Master Approximation Algorithm

This section describes a nonbipartite master approximation algorithm. Figure 6 illustrates the structure of a *nonbipartite latticoid*, which is formally defined as follows.

Definition 4 *A* nonbipartite latticoid, *$\hat{L}$, of a nonbipartite lattice L is an infinite graph that contains an infinite sequence of contact edges (a_i, b_i) with the following properties:*

– *There is a polynomial extensible path p_i^a from a_i to a_{i+1} ($\gamma = 1$),*

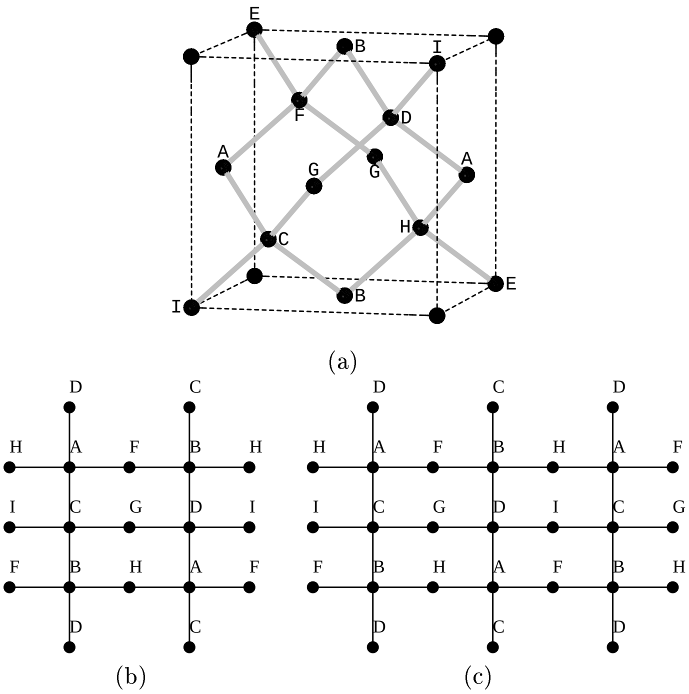


Figure 4: Embedding the $\hat{\mathbf{L}}_0^2$ latticoid into a diamond lattice: (a) labeled unit cell, (b) embedding onto plane of unit cells with embedded latticoid, and (c) extending this embedding.

– *There is a constant $\kappa > 0$ such that for every i and j, $d_{p_i^a}(a_i, a_{i+1}) = \kappa$, and*
– *The set of paths $\{p_i^a, p_{i+1}^a \mid i = 1, \ldots\}$ are mutually extensibly disjoint.*

The dilation *of the bipartite latticoid is $\Delta_{\hat{L}} = \kappa$.*

The nonbipartite master approximation algorithm places hydrophobics along the path of a_i's in such a manner that as few hydrophobics are placed outside the path as possible. Note that because the hydrophobic-hydrophobic contacts are constructed along a path, the extensible paths may lie on either side of this path.

For a nonbipartitate latticoid $\hat{L}$, the *dilation* $\Delta_{\hat{L}}$ is the minimal length of a path from a_i to a_{i+1}. Thus the nonbipartite master approximation algorithm guarantees that at least $\lfloor N(s)/\Delta_{\hat{L}} \rfloor$ hydrophobic amino acids lie along the path of a_i's. Given this, we can prove the following performance guarantee for a nonbipartite master approximation algorithm $\mathcal{B}$ on lattice L with latticoid $\hat{L}$.

Proposition 2 $R^\infty(\mathcal{B}_{\hat{L}}) \geq 2/(\Delta_{\hat{L}}(\delta(L) - 2))$.

Proof. We can bound the energy of the optimal conformation by $OPT(s) \geq -(\delta(L) - 2)N(s)/2 - 2$ since every hydrophobic has $\delta(L)$ neighbors that can

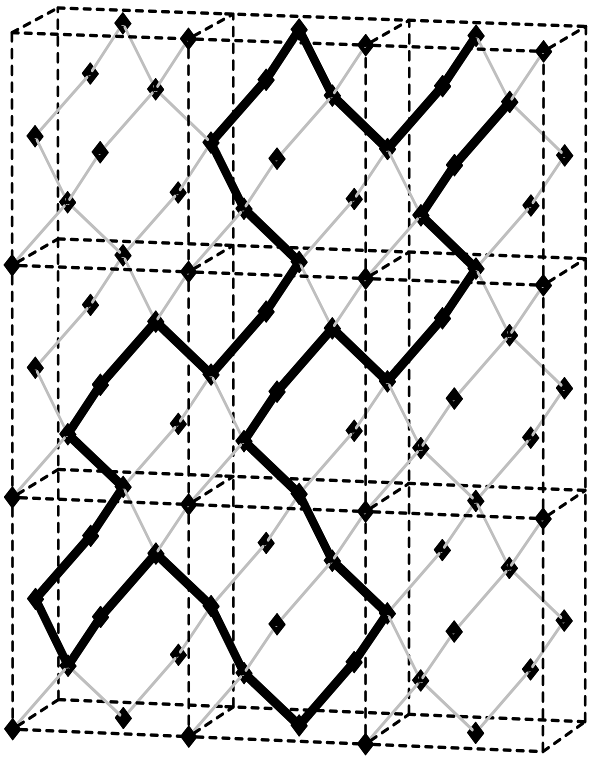


Figure 5: Illustration of the embedding of the bipartite latticoid $\hat{\mathbf{L}}_0^2$ into a diamond lattice.

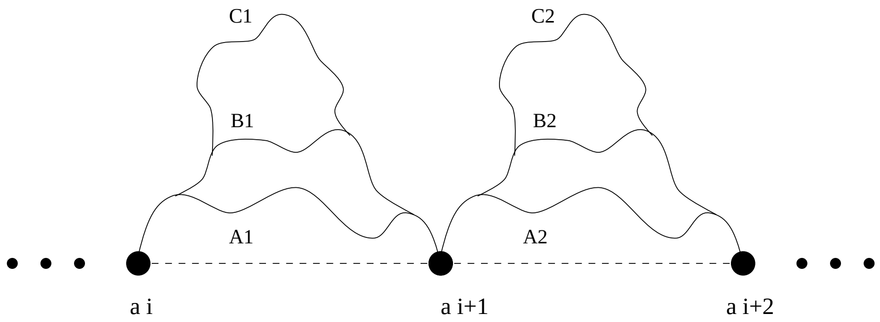


Figure 6: A symbolic illustration of the structure of nonbipartite latticoids.

form contacts. Now $B(s) \leq -\lfloor N(s)/\Delta_{\hat{L}} \rfloor + 1 \leq -N(s)/\Delta_{\hat{L}} + 2$. Thus bound on $R^\infty(\mathcal{B}_{\hat{L}})$ follows from Lemma 1.

4 A Complexity Theory for Protein Folding on Crystal Lattices

In this section we extend the methods used in the previous section to provide a framework for analyzing the design of efficient approximation algorithms with provable performance guarantees on lattices. The unifying theme is polynomial approximability asymptotic within a constant of optimal. This theory defines polynomial embedding reductions from one lattice to another, and relates the approximability on the first lattice to the approximability on the second. Further, this theory includes a notion of *completeness*, which defines the "hardest" members in the class.

A *core* of a lattice L is a set of latticoids $D(L) = \{\hat{L}^1, \hat{L}^2, \ldots\}$, where $D(L)$ is finite or countably infinite. We will use lattice cores to extend the role of the latticoid in our previous analysis. Specifically, a lattice core can contain multiple latticoids of the same lattice. For example, we could have $D(\mathbf{L}_0) = \{\hat{\mathbf{L}}_0^2, \hat{\mathbf{L}}_0^H\}$ from Figure 3.

Folding algorithms in a lattice L_1 can be transferred to folding algorithms in another lattice L_2, a folding "reduction", if the sublattice used in L_1 by the approximation algorithm can be embedded in L_2. Note that this reduction does not require that we explicitly embed the sublattice of L_1 in L_2. Instead, we simply need a polynomial algorithm for maping a specific conformation in L_1 into a corresponding conformation in L_2 that preserves an interesting set of hydrophobic contacts. However, this reduction does require that L_1 and L_2 be *consistent*, which means that either they are both bipartite or nonbipartite lattices. Consistency ensures that the bounds on the optimal conformation are similar for both lattices.

For example, consider the bipartite master approximation algorithm described in Section 3. To illustrate how this could be applied to the diamond lattice, we described how the $\hat{\mathbf{L}}_0^2$ latticoid can be embedded into the diamond lattice. However, it is not necessary to generate the conformations for this algorithm in the diamond lattice itself. Instead, we can generate conformations in the cubic lattice such that they are constrained to lie on the $\hat{\mathbf{L}}_0^2$ latticoid, and subsequently use the mapping graphically described in Figure 4c to construct a conformation in the diamond lattice.

Polynomial embeddings like this should be easy to construct because the unit cells in each lattice have a a finite description, and the symmetries in the crystal lattice are with respect to the neighboring cells (and thus also of finite description). Let 2^L refer to the set of sublattices of lattice L. This notion of reduction is formalized in the following definition.

Definition 5 *A* polynomial embedding reduction *of L_1 to L_2 via core $D(L_1)$ is a polynomial time function $\psi : 2^{\hat{L}_1} \to 2^{\hat{L}_2}$ such that*

1. L_1 and L_2 are consistent,
2. $\hat{L}_1 \in D(L_1)$,

3. *$\hat{L}_2$ is a sublattice of L_2,*
4. *$\psi(\hat{L}_1)$ is lattice isomorphic to $\hat{L}_2$ (i.e. graph isomorphic), and*
5. *the time complexity for mapping $\hat{L} \in 2^{\hat{L}_1}$ into $\hat{L}_2$ is polynomial in the number of vertices and edges of $\hat{L}$.*

If there is a polynomial embedding reduction from L_1 to L_2 via core $D(L_1)$, we write $L_1 \propto_{D(L_1)} L_2$.

Let $f_L(s) = X(s)$ if L is bipartite and $N(s)$ otherwise. We say that a lattice L is *polynomial kernel-approximable* if there is a polynomial algorithm $\mathcal{A}$ and constants $\alpha_L, \beta_L \in \mathbf{Z}^{>0}$ such that for all protein instances s, $A(s) \leq -\alpha_L f_L(s) + \beta_L$. This type of approximability reflects the energetic guarantees provided by all of the approximation methods that have been described in the literature. Consequently, we describe the square, cubic, triangular and face-centered cubic lattices as polynomial kernal-approximable [1, 7, 12]. We say that a class of lattices $\mathcal{L}$ is *polynomial kernel-approximable* if for every $L \in \mathcal{L}$, L is polynomial kernel-approximable, and let **PKAL** be the class of polynomial kernel-approximable lattices. From Lemma 1 and the fact that the vertex degree in lattices is finite, it follows that $\forall L \in \mathbf{PKAL}$ there exists a constant $\tau_L > 0$ such that $R^\infty(\mathcal{A}) \geq \tau_L$.

Now consider a lattice L with core $D(L)$. We say that L is *polynomially core kernal-approximable* if for all $\hat{L} in D(L)$, L is polynomial kernal-approximable with an algorithm $\mathcal{A}_{\hat{L}}$ that generate conformations strictly on $\hat{L}$. If L is polynomially core kernal-approximable then clearly $L \in \mathbf{PKAL}$. The following lemma shows how a lattice core can be used to ensure the approximability via a reduction.

Lemma 4. *Consider L_1 with core $D(L_1)$ that is polynomially core kernal-approximable. If $L_1 \propto_{D(L_1)} L_2$, then $L_2 \in \mathbf{PKAL}$.*

Proof. If $L_1 \propto_{D(L_1)} L_2$ then there exists a sublattice $\hat{L}_2$ that is graph isomorphic to a sublattice $\hat{L}_1 \in D(L_1)$. Since L_1 is polynomially core kernal-approximable there exists an approximation algorithm $\mathcal{Z}$ for $\hat{L}_1$ such that $\mathcal{Z}(s) \leq -\alpha_L f_L(s) + \beta_L$ for constants α_L and β_L. Now consider an approximation algorithm $\mathcal{Y}$ that applies algorithm $\mathcal{Z}$ to an instance s, and then applies the reduction to map the conformation in $\hat{L}_1$ to a conformation in $\hat{L}_2$. Clearly, $\mathcal{Y}(s) = \mathcal{Z}(s)$, so algorithm $\mathcal{Y}$ generates conformations in L_2 such that $\mathcal{Y}(s) \leq -\alpha_L f_L(s) + \beta_L$. By the consistency of the reduction, it follows that $L_2 \in \mathbf{PKAL}$.

The central concept of the complexity theory is the notion of completness defined as follows.

Definition 6 *Let $\mathcal{L}$ be a class of lattices. A lattice L is called $\mathcal{L}$-*complete via core $D(L)$ *if*

1. *$L \in \mathcal{L}$, and*
2. *$\forall L' \in \mathcal{L}$, $L \propto_{D(L)} L'$.*

Similar to the theory of NP-completness, if any member of the complete set is core-approximable then we can design polynomial approximation algorithms for all lattices in the class.

Theorem 1 *Let L be a lattice with core $D(L)$. If L is $\mathcal{L}$-complete and polynomially core kernal approximable, then $\mathcal{L} \subseteq$ **PKAL**.*

Proof. Consider an arbitrary $L' \in \mathcal{L}$. Since L is polynomially core kernal-approximable, from Lemma 4 we know that the fact that $L \propto_{D(L)} L'$ implies that $L' \in$ **PKAL**. Since this applies to all L', $\mathcal{L} \subseteq$ **PKAL**.

5 Approximable Lattices for the HP Model

In this section we describe a class of lattices $\mathcal{L}$ for which performance guaranteed approximation algorithms exist. $\mathcal{L}$ is a broad class of lattices that includes many of the lattices previously used in lattice models for protein folding. This class of lattices is divided into bipartite and nonbipartite lattices, which we describe seperately. See Sands [18] and Wells [23] for further details on many of the lattices that we describe below.

Consider the square lattice L and the core $D(L) = \{\hat{\mathbf{L}}_0^2, \hat{\mathbf{L}}_0^H\}$. We can apply the bipartite master approximation algorithm to show that L is polynomially core kernal approximable. We now describe a class of lattices $\mathcal{L}$ for which L is $\mathcal{L}$-complete via $D(L)$:

- **Square and Cubic Lattices**: The square lattice is clearly a sublattice of the three-dimensional cubic lattice. Further, the square lattice can be simply embedded into Bravais lattices like the triclinic and triagonal lattices [18], which simply rescale and shift the angles of the cubic lattice (e.g. see Figure 7(a)).
- **Diamond and Flourite Lattices**: In Section 3.2 we saw how $\hat{\mathbf{L}}_0^2$ could be embedded in the diamon lattice. Figure 7(b) shows a flourite lattice structure, for which the diamond lattice is a sublattice. It follows that the square lattice has a polynomial embedding reduction into the flourite lattice via $D(L)$.
- **Generalized Cartesian Lattices**: Several researchers have considered generalized lattices that take points from the square or cubic lattice but defined a generalized neighborhood for edges and contacts. For example, Figure 7(c) shows the neighborhood for the "210 lattice" that Skolnick and Kolinkski [21] use to place α-carbons. In this lattice, the α-carbons are connected by the 3D generalization of the "knight's walk" in chess.
  Now consider a lattice L' formed with a symmetric generalized neighborhood. Let (a, b) represent a neighborhood move on this lattice. From the symmetry of the neighborhood structure, it follows that (b, a), $(-a, -b)$, and $(-b, -a)$ are also neighbors. To realize the embedding of the cubic lattice in L', we equate the edge $(1, 0)$ with (a, b) and the edge $(0, 1)$ with (b, a). Using these vectors as a basis, the integral combinations of them form a cubic lattice. Figure 7 illustrates this embedding for the "210 lattice".
  Note that a generalized basis can generate either a nonbipartite or bipartite lattice of edges. If the lattice is bipartite, then this reduction suffices to show that the square lattice has a polynomial embedding reduction into the generalied Cartesian lattice L'.
- **Hexagonal Lattice**: The latticoid $\hat{\mathbf{L}}_0^H$ can be embedded into the hexagonal lattice by noting that the extensible paths are generated shifting the

initial path around adjacent hexagons. Figure 8 illustrates the extension of a path between two hydrophobics that form the hydrophobic core of the conformation. This type of extension is easy to generate, so the reduction is polynomial. The catalog of lattices in Wells [23] contains many bipartite three-dimensional lattices into which the hexagonal lattice can be embedded. It follows that the square lattice has a polynomial embedding reduction into all of these lattices.

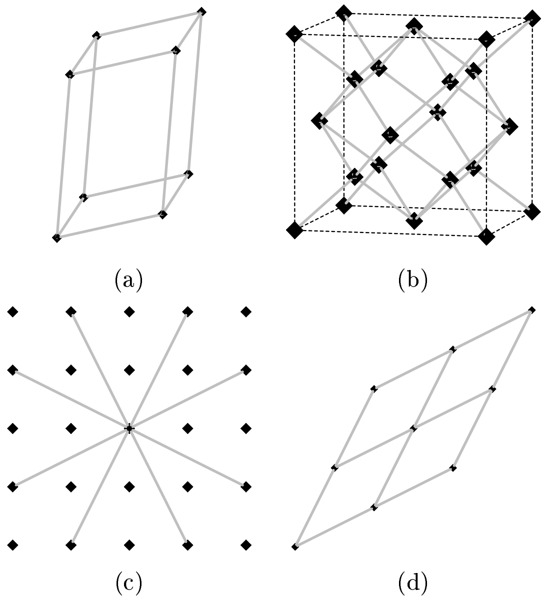


Figure 7: Bipartite crystal lattices: (a) triclinic, (b) flourite, (c) the "210 lattice", and (d) the embedding of L into the "210 lattice".

We have shown that the square lattice has a polynomial embedding reduction into all of these lattices using $D(L)$. Thus L is $\mathcal{L}$-complete via $D(L)$. Since L is polynomially core kernal-approximable, it follows that $\mathcal{L} \subset$ **PKAL**.

5.1 Nonbipartite Lattices

Consider the triangular lattice $\bar{L}$ and the core $D(\bar{L}) = \{\bar{\mathbf{L}}_0\}$ illustrated in Figure 9(a). We can apply the nonbipartite master approximation algorithm to show that $\bar{L}$ is polynomially core kernal approximable. We now describe a class of lattices $\mathcal{L}'$ for which $\bar{L}$ is $\mathcal{L}$-complete via $D(\bar{L})$:

– **Face Centered Cubic**: Consider a single plane of faces for a face centered cubic lattice. Figure 9(b) illustrates how $\bar{\mathbf{L}}_0$ can be embedded on this face.

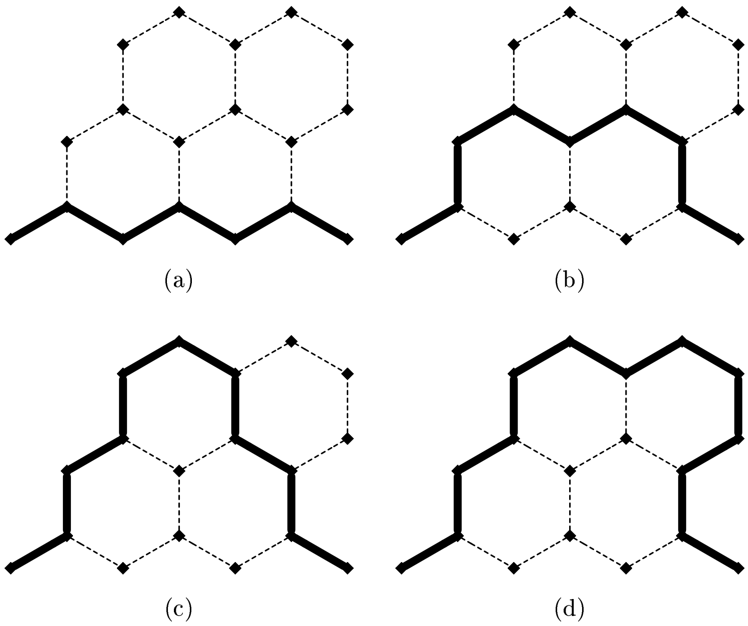
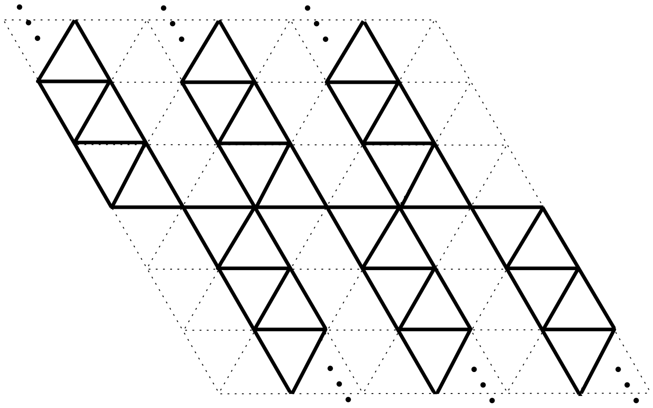


Figure 8: Illustration of the embedding of extensible paths from $\hat{\mathbf{L}}_0^H$ into the hexagonal lattice. Figure (a) shows the initial path that is extended by two in (b), (c) and (d).

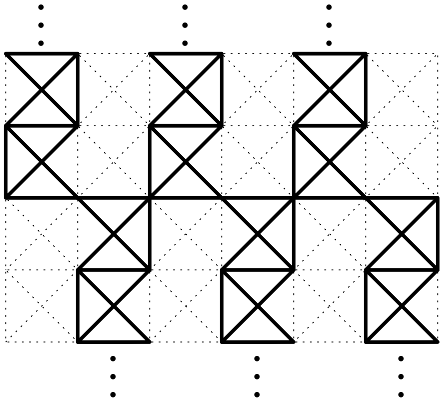
Thus the triangular lattice has a polynomial embedding reduction into the face centered cubic lattice.

- **Body Centered Cubic**: Consider a single plane of faces for a face centered cubic lattice. On top and behind each square of points lies a point that is in contact with each point on the square. This sublattice of edges has the same connectivity structure as a single plane of faces for the face centered cubic lattice. Consequently, the triangular lattice has a polynomial embedding reduction into the body centered cubic lattice.
- **3D Close Packed**: Close packed lattices are composed of layers of 2D close packed lattices. These layers can be put in contact in several different ways, providing an infinite number of possible close packings in 3D. The 2D close packed lattice structure is simply the triangular lattice structure, so these lattices are polynomially core kernal approximable.

We have shown that the triangular lattice has a polynomial embedding reduction into all of these lattices using $D(\bar{L})$. Thus $\bar{L}$ is $\mathcal{L}'$-complete via $D(\bar{L})$. Since $\bar{L}$ is polynomially core kernal-approximable, it follows that $\mathcal{L}' \subset \mathbf{PKAL}$.



(a)



(b)

Figure 9: Illustration of (a) the nonbipartite latticoid used by the nonbipartite master approximation algorithm in L', the triangular lattice. This latticoid can be embedded in the faced centered cubic lattice as shown in (b).

6 Discussion

We have described master approximation algorithms for bipartite and nonbipartite lattices that illustrate how performance guaranteed approximation algorithms can be applied to a wide range of crystal lattices. The general applicability of these master approximation algorithms is limited to graphs for which latticoid subgraphs can be efficiently embedded. Consequently, these results fall short of demonstrating that performance guaranteed approximability is an algorithmic invariant for crystal lattices.

However, the classes of lattices described in the previous section, $\mathcal{L}$ and $\mathcal{L}'$ include a wide range of lattices that have played a significant role in the analysis of protein structure prediction. Although the master approximation algorithms do not necessarily provide the best provable performance guarantees in all cases, their applicability to such a broad range of well-studied lattices does indicate that there is some measure of lattice independence for reasonable lattice graphs. This suggests that the algorithmic mechanisms used to generate these approximate conformations may play a role in biological systems.

Although our analysis has focused on simple chain models, we expect that it can be simply generalized to more structured protein models. For example, Hart and Istrail [15] and Heun [16] describe performance guaranteed approximation algorithms for a side-chain lattice model. These results are applicable to square, cubic, face-centered cubic and extended cubic lattice models. We conjecture that these results can, in fact, be similarly be extended to a broader range of well-studied lattice models.

Acknowledgements

Our thanks to Ken Dill for suggesting the extension of our previous results to other lattice models and for discussions that inspired this work. This work was supported by the Mathematics, Information and Computational Science program, U.S. Department of Energy, Office of Energy Research. This work was performed at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

References

[1] Richa Agarwala, Serafim Batzogloa, V. Dančík, Scott E. Decatur, S. Hannenhalli, M. Farach, S. Muthukrishnan, and S. Skiena. Local rules for protein folding on a triangular lattice and generalized hydrophobicity for the HP model. In *Proc 8th Symp Discrete Algorithms*, pages 390–399, 1997.

[2] Neil W. Ashcroft and N. David Mermin. *Solid State Physics*. Holt, Rinehart and Winston, 1976.

[3] Jonathan Atkins and William E. Hart. On the intractability of protein folding with a finite alphabet of amino acids. *Algorithmica*, 25:279–294, 1999.

[4] B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *J Comp Bio*, 5(1):27–40, 1998.

[5] D. G. Covell and R. L. Jernigan. *Biochemistry*, 29:3287, 1990.

[6] P Crescenzi, D Goldman, C Papadimitriou, A Piccolboni, and M Yannakakis. On the complexity of protein folding. *J Comp Bio*, 5(3), 1998.

[7] Vlado Dančík and Sridhar Hannenhalli. Protein folding on a triangular mesh, May 1996. Unpublished research.

[8] Ken A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24:1501, 1985.

[9] Ken A. Dill, Sarina Bromberg, Kaizhi Yue, Klaus M. Fiebig, David P. Yee, Paul D. Thomas, and Hue Sun Chan. Principles of protein folding: A perspective from simple exact models. *Prot. Sci.*, 4:561–602, 1995.

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

[11] A. M. Gutin and E. I. Shakhnovich. Ground state of random copolymers and the discrete random energy model. *J. Chem. Phys.*, 98:8174–8177, 1993.

[12] William Hart and Sorin Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal. *Journal of Computational Biology*, 3(1):53–96, 1996.

[13] William Hart and Sorin Istrail. Robust proofs of NP-hardness for protein folding: General lattices and energy potentials. *Journal of Computational Biology*, 4(1):1–20, 1997.

[14] William E. Hart and Sorin Istrail. Invariant patterns in crystal lattices: Implications for protein folding algorithms. In *Combinatorial Pattern Matching*, Lecture Notes in Computer Science 1075, pages 288–303, New York, 1996. Springer.

[15] William E. Hart and Sorin 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):241–259, 1997.

[16] Volker Heun. Approximate protein folding in the HP side chain model on extended cubic lattices. In *Proc 7th Annual European Symp on Algorithms*, volume 1643 of *Lecture Notes in Computer Science*, pages 212–223. Springer-Verlag, 1999.

[17] Martin Karplus and Eugene Shakhnovich. *Protein folding: Theoretical studies of thermodynamics and dynamics*, chapter 4, pages 127–195. W. H. Freeman and Company, 1993.

[18] Donald E. Sands. *Introduction to Crystallography.* Dover Publications, Inc., New York, 1975.

[19] 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.

[20] Andrzej Sikorski and Jeffrey Skolnick. Dynamic Monte Carlo simulations of globular protein folding/unfolding pathways. II. α-helical motifs. *J. Molecular Biology*, 212:819–836, July 1990.

[21] Jeffrey Skolnick and Andrzej Kolinski. Simulations of the folding of a globular protien. *Science*, 250:1121–1125, 1990.

[22] R. Unger and J. Moult. Finding the lowest free energy conformation of a protein is a NP-hard problem: Proof and implications. *Bull. Math. Bio.*, 55(6):1183–1198, 1993.

[23] A. F. Wells. *Three-dimensional nets and polyhedra.* American Crystallographic Association, 1979.