Optimally Confining Lattice Polymers

Robert D. Barish* Tetsuo Shibuya†

Abstract

We introduce the Lattice Polymer Confinement Problem (LPCP), where provided a graph G corresponding to a solid or hole-containing finite lattice, and provided a finite set of vertex-wise lengths $\mathcal{L} \subset \mathbb{N}$ of lattice polymers modeled as Self-Avoiding Walks (SAWs), the objective is to delete the fewest possible number of vertices in G to satisfy a bound $\mathcal{S}_{(G,\mathcal{L})} \leq \Omega$ on a sum over the configuration entropies of each polymer. In this context, we use Boltzmann's expression $\mathcal{S}_{(G,\mathcal{L})} = k_B \cdot \ln(\mathcal{W}+1)$ for the system configuration entropy, where $k_B \approx 1.380649 \cdot 10^{-23}\ J \cdot K^{-1}$ is Boltzmann's constant, and $\mathcal{W}$ corresponds to a sum over the number of SAWs modeling lattice polymers in a specified host graph. We also propose a novel Self-Avoiding Walk (SAW) centrality measure, $\mathcal{C}_{SAW}(\mathcal{L}, v_i)$, for a vertex v_i in a lattice or graph as a variation on the standard notion of betweenness centrality, which for a specified finite set $\mathcal{L} \subset \mathbb{N}$, corresponds to the fraction of length $l_i \in \mathcal{L}$ SAWs that cover v_i.

Letting G be an input lattice or graph for LPCP with vertex set V_G and edge set E_G, we show that LPCP is NP-hard as well as APX-hard $\forall \Omega \geq 0$ and for all finite $\mathcal{L} \subset \mathbb{N}_{\geq 2}$. On the other hand, letting $tr(G)$ be the treewidth of G, letting $\zeta_{tw} = f(tr(G)) \cdot \mathcal{O}(|V_G| + |E_G|)$ for some computable function f, and letting $\mathcal{Y}_{init}$ be the initial system configuration entropy, we prove the existence of an $\mathcal{O}\left(\zeta_{tw} \cdot |V_G|^3 \cdot \ln(|V_G|)\right)$ time $\left(\ln\left(e^{(\mathcal{Y}_{init})} - e^{\Omega}\right)\right)$-approximation algorithm for LPCP. We moreover establish that an $\mathcal{O}(\Psi)$ deterministic algorithm for SAW centrality with multiplicative error $1 \pm \epsilon$, which we remark can be derived from existing PTAS algorithms for counting bounded-length SAWs in graphs, correspondingly implies the existence of an $\mathcal{O}\left(\Psi \cdot |V_G|^3 \cdot \ln(|V_G|)\right)$ time $\left(\frac{\ln\left(e^{(\mathcal{Y}_{init})} - e^{\Omega}\right)}{1-2\epsilon}\right)$-approximation algorithm for LPCP.

Finally, we briefly analyze variations on LPCP, including a variant where we delete edges in lieu of vertices, and variant with "rigid" lattice polymers (e.g., lattice proteins) where every embedding must satisfy a set of consecutive dihedral angles for adjacent bonds.

1 Introduction

We introduce and analyze what we denote the Lattice Polymer Confinement Problem (LPCP), which concerns minimally modifying a solid or hole-containing finite lattice G such that, provided a finite set of vertex-wise lengths $\mathcal{L} \subset \mathbb{N}$ of lattice polymers modeled as Self-Avoiding Walks (SAWs), the system configuration entropy $\mathcal{S}_{(G,\mathcal{L})} = k_B \cdot \ln(\mathcal{W}+1)$ falls below a specified threshold $\Omega \geq 0$. In this context, $k_B \approx 1.380649 \cdot 10^{-23}\ J \cdot K^{-1}$ is Boltzmann's constant, $\mathcal{W}$ corresponds to a sum over the number of SAWs (modeling lattice polymers) of length $l_i \in \mathcal{L}$ in a specified host graph, and Ω should everywhere be assumed to have units $J \cdot K^{-1}$. Briefly, we can observe that the system configuration entropy is equivalently expressed as $\mathcal{S}_{(G,\mathcal{L})} = -k_B \cdot \sum_{i=1}^{\mathcal{W}} \left(\left(\frac{1}{\mathcal{W}+1}\right) \cdot \ln\left(\frac{1}{\mathcal{W}+1}\right)\right)$, allowing us to obtain the expression for Shannon entropy by substituting k_B with the reciprocal of the logarithm of the number of observed events for a discrete random process and $\left(\frac{1}{\mathcal{W}+1}\right)$ with the probability of a specific event.

We remark that such lattice polymer models have extensive precedence in the field of protein structure prediction and engineering [14, 22]. For illustrative examples, we refer the reader to Fig. 1 and Fig. 2, where we show LatFit [23, 24] generated (semi-rigid) embeddings of the peptide backbones for the NMR solution structure of an ShK potassium channel inhibitor toxin from sea anemone (PDB ID: 1ROO) on a $\mathbb{Z}^2$ integer lattice, and the crystal structure of an antifreeze protein from notched-fin eelpout (PDB ID: 5XQN) on a 210 "knight's tour" lattice, respectively.

Our inspiration for LPCP is a visually stunning experimental demonstration by Turner et. al. [26] of how entropy gradients can invoke forces on polymers. To briefly describe their experiment, Turner et. al. [26] began by manufacturing a microfluidic cell with two adjacent quasi-two-dimensional volumes, which we will refer to as J_{open} and $J_{pillars}$, where J_{open} is an otherwise open volume and $J_{pillars}$ is populated with $\approx 35\ nm$ diameter pillars with a $\approx 160\ nm$ center-to-center spacing. The authors then used an electric field to drag double-stranded T2-phage genomic DNA (having a contour length of $\approx 51\ \mu m$) from J_{open} to $J_{pillars}$, signifi-

*Division of Medical Data Informatics, Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan, rbarish@ims.u-tokyo.ac.jp

†Division of Medical Data Informatics, Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan, tshibuya@hgc.jp

cantly restricting the polymer's configuration freedom. Once the electric field was lifted and one end of a given polymer diffused into J_{open}, a CCD camera was used to observe the remainder of the molecule rapidly "recoiling" out of $J_{pillars}$, acting against a hydrodynamic drag with a force of $\approx 5.7\ fN$. The authors then determined that this force was driven almost entirely by a configuration entropy gradient, noted that it was within an order of magnitude of the $\approx 40\ fN$ force expected by a $\approx \Delta 1\ kB$ change in configuration entropy per polymer Kuhn length (i.e., length units of a semi-rigid polymer that can be approximated as segments of a freely-jointed chain [16]) moving from $J_{pillars}$ to J_{open}.

We now ask the question: if we treat the configuration entropy $\mathcal{S}_{(G,\mathcal{L})}$ for one or more lattice polymers embedded in a solid lattice as roughly equivalent to the embedding of real polymers in an open volume akin to J_{open}, how can we minimally modify the lattice (e.g., by deleting vertices) to create a volume akin to $J_{pillars}$? Here, the aforementioned LPCP problem, which we formally define below, represents our attempt to formalize and generalize this problem.

Definition 1 *Lattice Polymer Confinement Problem,* $LPCP\left(G, \mathcal{L}, \Omega\right)$

Input: A graph G with vertex set V_G, corresponding to a solid or hole-containing finite lattice, a finite set of vertex-wise lengths $\mathcal{L} \subset \mathbb{N}$ of lattice polymers modeled as Self-Avoiding Walks (SAWs), and an upperbound Ω for the configuration entropy $\mathcal{S}_{(G,\mathcal{L})}$ of the system. Here, $\mathcal{S}_{(G,\mathcal{L})} = k_B \cdot \ln\left(\mathcal{W}+1\right)$, where $k_B \approx 1.380649 \cdot 10^{-23} J \cdot K^{-1}$ is Boltzmann's constant, and $\mathcal{W}$ corresponds to a sum over the number of embeddings in G of each lattice polymer corresponding to a SAW of length $l_i \in \mathcal{L}$.
Objective: Return a minimum cardinality set of vertices $\mathcal{Q} \subseteq V_G$ whose deletion converts G into a graph G' where we have that $\mathcal{S}_{(G',\mathcal{L})} \leq \Omega$.

For illustrative examples of $LPCP\left(G, \mathcal{L}, \Omega\right)$ and what (approximate) witnesses look like, we refer the reader to Fig. 3, where we show instances of input graphs G corresponding to: (a) a 6×6 induced subgraph of a $\mathbb{Z}^2$ integer lattice; (b) a $3 \times 3 \times 3$ induced subgraph of a $\mathbb{Z}^3$ integer lattice; (c) an induced subgraph of a triangular lattice; and (d) an induced subgraph of a honeycomb lattice. In each of the examples from Fig. 3(a–d), we also show a set of (white) vertices that would be selected in the specified order (first 1, then 2, etc.) for deletion by a greedy algorithm attempting to minimize $\mathcal{S}_{(G,\mathcal{L})}$. In the Fig. 3(e) table, we show the approximate configuration entropy for the examples in Fig. 3(a–d) (recall that $k_B \approx 1.380649 \cdot 10^{-23} J \cdot K^{-1}$), as well as the configuration entropies following each vertex deletion.

As a subroutine of our greedy algorithms for the LPCP problem, we also introduce a novel Self-Avoiding Walk (SAW) vertex centrality measure as a variation on *betweenness centrality*. This measure assigns a score to the vertices of a simple undirected graph based on the fraction of all possible SAW embeddings of specified lengths $l_i \in \mathcal{L}$ they are covered by, and accordingly allows one to rank vertices in a graph according to the effect of their deletion on the system configuration entropy. More specifically, letting G be a simple graph with vertex set V_G, and letting $f_{(SP,all)}\left(G, v_a, v_b\right)$ and $f_{(SP,v_i)}\left(G, v_a, v_b\right)$ be functions which return the number of shortest paths from a vertex $v_a \in V_G$ to a vertex $v_b \in V_G$ and the number of such paths traversing the vertex $v_i \notin \{v_a, v_b\}$, respectively, we can recall that the *betweenness centrality* [17, 18, 27] for a vertex $v_i \in V_G$ is given by $\mathcal{C}_{Betweenness}\left(G, v_i\right) = \sum_{(a,b \in [1,|V_G|] \wedge a<b \wedge a \neq i \wedge b \neq i)} \left\{ \begin{pmatrix} \frac{f_{(SP,v_i)}(G,v_a,v_b)}{f_{(SP,all)}(G,v_a,v_b)} \end{pmatrix}, \quad f_{(SP,all)}\left(G, v_a, v_b\right) \neq 0 \\ 0, \quad f_{(SP,all)}\left(G, v_a, v_b\right) = 0 \right\}$.
Now, letting G and V_G be defined as before, and letting $f_{(SAW,all)}\left(G, \mathcal{L}\right)$ be a function which returns the number of all simple paths (equiv. SAWs) of all possible vertex-wise lengths $l_i \in \mathcal{L}$ in G, we can define the *SAW centrality* for a vertex $v_i \in V_G$ as $\mathcal{C}_{SAW}\left(G, \mathcal{L}, v_i\right) = \left\{ \begin{pmatrix} \frac{f_{(SAW,all)}(G-v_i,\mathcal{L})}{f_{(SAW,all)}(G,\mathcal{L})} \end{pmatrix}, \quad \text{for } f_{(SAW,all)}\left(G, \mathcal{L}\right) \neq 0 \\ 0, \quad \text{for } f_{(SAW,all)}\left(G, \mathcal{L}\right) = 0 \right\}$.

To begin our analysis of LPCP, we first establish hardness results. In particular, we show that LPCP is NP-hard even if G is a subgraph of a $\mathbb{Z}^2$ integer lattice and we have either the constraint that $|\mathcal{L}| = 1$ or the constraint that $\mathcal{L} = \{1, 2, \ldots, |V_G|\}$ (Proposition 1). If G is allowed to be an arbitrary simple undirected graph, we moreover show that LPCP is NP-hard as well as APX-hard $\forall \Omega \geq 0$ and for all finite $\mathcal{L} \subset \mathbb{N}_{\geq 2}$ (Proposition 2).

We next detail approximation algorithms for LPCP. In particular, letting $tr\left(G\right)$ be the treewidth of G, letting $\zeta_{tw} = f\left(tr\left(G\right)\right) \cdot \mathcal{O}\left(|V_G| + |E_G|\right)$ for some computable function f, and letting $\mathcal{Y}_{init}$ be the initial system configuration entropy, we prove the existence of an $\mathcal{O}\left(\zeta_{tw} \cdot |V_G|^3 \cdot \ln\left(|V_G|\right)\right)$ time $\left(\ln\left(e^{(\mathcal{Y}_{init})} - e^{\Omega}\right)\right)$-approximation algorithm (Theorem 3). We additionally show that an $\mathcal{O}\left(\Psi\right)$ deterministic algorithm for SAW centrality with multiplicative error $1 \pm \epsilon$ correspondingly implies the existence of an $\mathcal{O}\left(\Psi \cdot |V_G|^3 \cdot \ln\left(|V_G|\right)\right)$ time $\left(\frac{\ln\left(e^{(\mathcal{Y}_{init})} - e^{\Omega}\right)}{1-2\epsilon}\right)$-approximation algorithm (Theorem 5).

Finally, we show how the aforementioned approximation algorithms extend to variations on LPCP where we delete edges in lieu of vertices (Corollary 8), as well as a variant where we consider the configuration entropies of "rigid" lattice polymers (e.g., lattice proteins) akin to those shown in Fig. 1 and Fig. 2 (Remark 1).

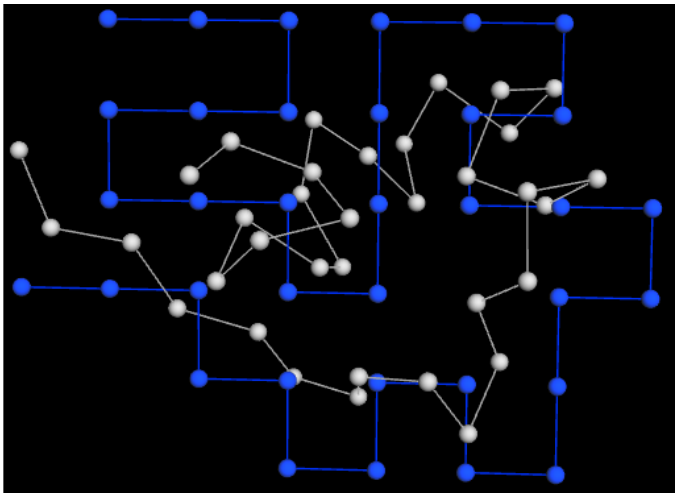


Figure 1: LatFit [23, 24] generated $\mathbb{Z}^2$ integer lattice embedding of the backbone for the NMR solution structure of an ShK potassium channel inhibitor toxin from sea anemone (PDB ID: 1ROO); the lattice embedding of the protein backbone is illustrated with (blue) vertices and edges, and the original structure of the protein backbone is illustrated with (white) vertices and edges.

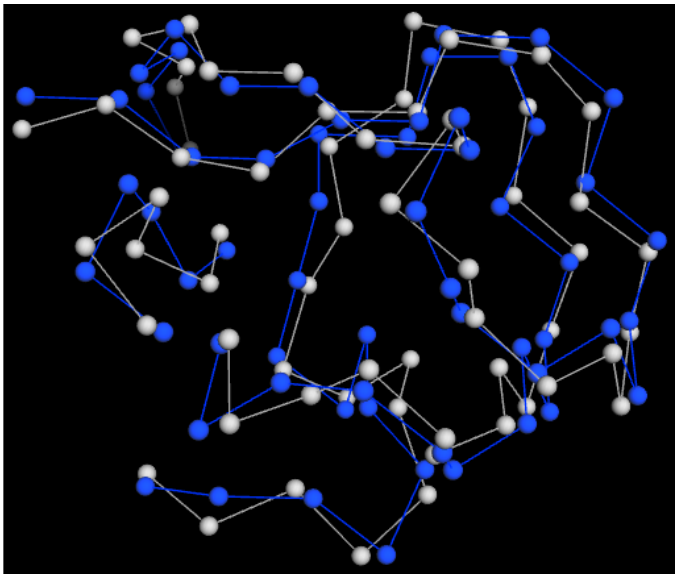
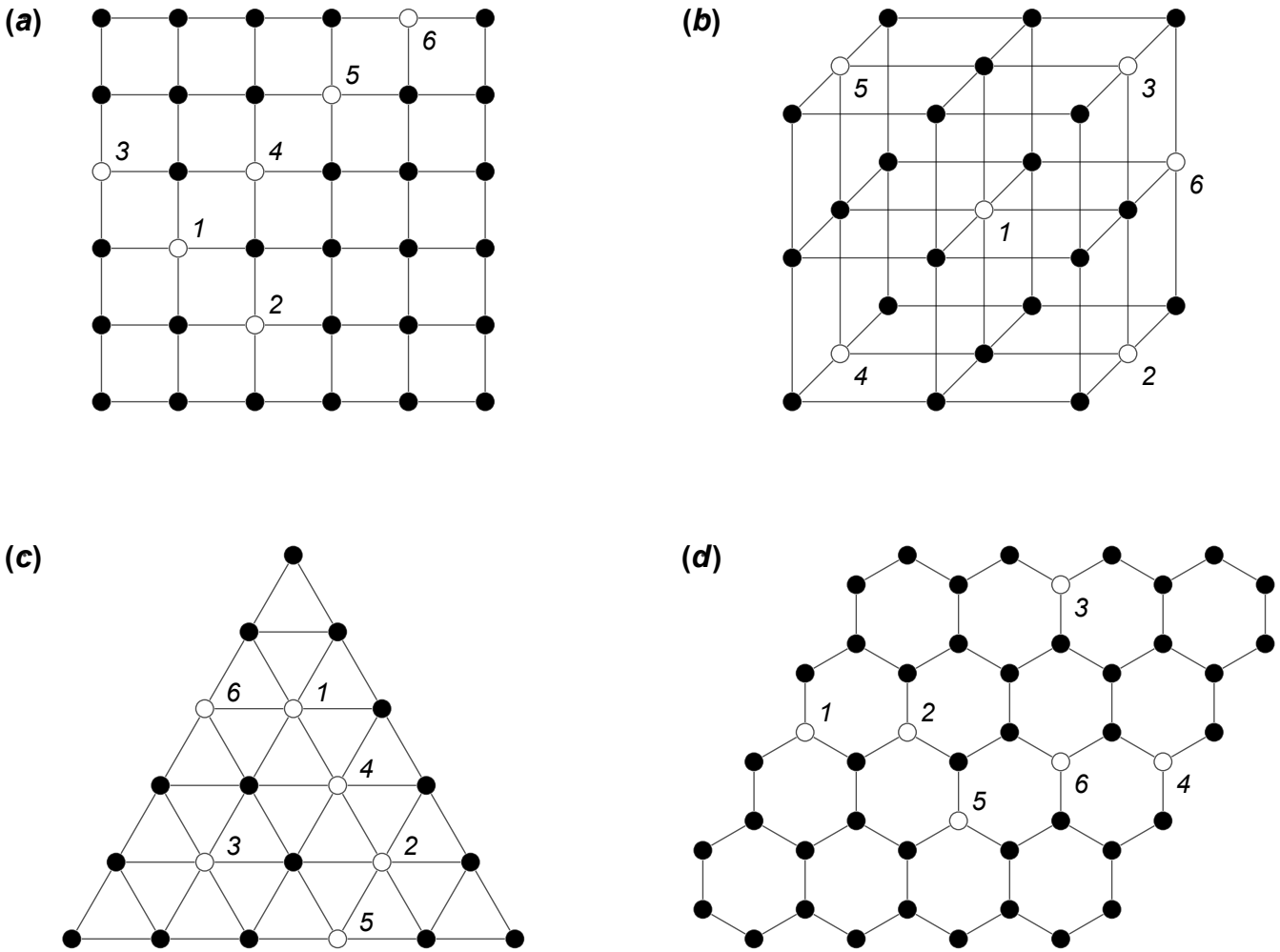


Figure 2: LatFit [23, 24] generated 210 "knight's tour" lattice embedding of the backbone for the crystal structure (obtained via X-ray diffraction techniques) of an antifreeze protein from notched-fin eelpout (PDB ID: 5XQN); the lattice embedding of the protein backbone is illustrated with (blue) vertices and edges, and the original structure of the protein backbone is illustrated with (white) vertices and edges.

2 Preliminaries

2.1 Graph theoretic terminology

We will generally follow definitions that are more-or-less standard (see, e.g., Diestel [11]). However, for some brief clarifications, when we use the term *graph* we are everywhere referring to simple undirected and unweighted graphs. We call a graph *cubic* if and only if all of its vertex degrees are uniformly equal to 3, and *subcubic* if and only if it has maximum vertex degree 3. Concerning paths and cycles in graphs, a path or cycle is called *simple*, or a Self-Avoiding Walk (SAW) in the case of paths, if it does not revisit either edges or vertices, called *Hamiltonian* if it is simple and covers all vertices, and called *induced* if it is also an induced subgraph. Here, the Hamiltonian cycle problem is the problem of deciding the existence of a Hamiltonian cycle in a graph, and the *st*-path problem and *st*-Hamiltonian



(e)

	Deleted Vertices:	None	{1}	{1,2}	{1,2,3}	{1,2,3,4}	{1,2,3,4,5}	{1,2,3,4,5,6}
(a)	6×6 Integer Lattice, $\mathcal{S}_{(G,\mathcal{L})}\,(J{\cdot}K^{-1}) \cdot (k_B)^{-1}$:	19.8946	18.2877	16.5245	14.7857	13.0960	11.2828	9.36888
(b)	3×3×3 Integer Lattice, $\mathcal{S}_{(G,\mathcal{L})}\,(J{\cdot}K^{-1}) \cdot (k_B)^{-1}$:	20.1267	18.0250	16.3276	14.5983	12.8702	11.1030	9.48562
(c)	Triangular Lattice, $\mathcal{S}_{(G,\mathcal{L})}\,(J{\cdot}K^{-1}) \cdot (k_B)^{-1}$:	16.3800	14.2945	12.1922	10.0761	8.51759	6.90174	4.95583
(d)	Honeycomb Lattice, $\mathcal{S}_{(G,\mathcal{L})}\,(J{\cdot}K^{-1}) \cdot (k_B)^{-1}$:	16.5431	15.0823	13.5212	11.9914	10.5050	9.06439	7.23778

Figure 3: Illustrative examples of $LPCP\,(G, \mathcal{L}, \Omega)$ problem instances, where letting $\mathcal{L}$ be the set of all possible SAW lengths, for each graph G in **(a–d)** the order of the first 6 vertices (colored white) selected by a naive greedy algorithm minimizing the system configuration entropy is shown (with the labels "1" for the first selected vertex, "2" for the second selected vertex, etc.). The approximate system configuration entropies before and after each successive vertex deletion event, divided by k_B, are given in the table shown in **(e)**. Here, **(a)** corresponds to a 6×6 induced subgraph of a $\mathbb{Z}^2$ integer lattice (36 vertices and 60 edges), **(b)** corresponds to a $3 \times 3 \times 3$ induced subgraph of a $\mathbb{Z}^3$ integer lattice (27 vertices and 54 edges), **(c)** corresponds to an induced subgraph of a triangular lattice (21 vertices and 45 edges), and **(d)** corresponds to an induced subgraph of a honeycomb lattice (48 vertices and 63 edges).

path problem is the problem of deciding the existence of a simple path (equiv., SAW) and Hamiltonian path, respectively, between a pair of vertices v_s and v_t.

2.2 Fixed-parameter tractability and intractability

A problem can be denoted *Fixed-Parameter Tractable* (FPT) if, letting x be a string encoding a given prob-

lem instance and $f(k)$ be any computable function, its time complexity can be written as $f(k) \cdot |x|^{\mathcal{O}(1)}$. With regard to parameterized hardness, we concern ourselves with completeness for the class $W[1]$ of all parameterized languages that can be encoded as Boolean decision circuits with *weft* at most 1 (see, e.g., ref. [13]). Here, a circuit with weft k can have at most k large gates (i.e. degree ≥ 3 vertices in the finite directed acyclic graph representation of the circuit) along any given path from an input node to an output node.

2.3 Approximation tractability and intractability

Concerning approximation tractability, we concern ourselves with the notion of a Polynomial-Time Approximation Scheme (PTAS) and Fully Polynomial-Time Approximation Scheme (FPTAS). Here, for some error parameter $\epsilon > 0$, a PTAS is a deterministic algorithm which produces a solution for a given optimization problem with a multiplicative error of $1 \pm \epsilon$ (typically $1-\epsilon$ and $1+\epsilon$ for maximization and minimization problems, respectively), with a running time polynomial in length of an input string specifying the optimization problem. If a PTAS also has a running time polynomial in $\frac{1}{\epsilon}$, then we refer to the PTAS as a FPTAS. With regard to approximation hardness, we concern ourselves with hardness for the class APX of problems admitting a constant-ratio approximation algorithm. As there are problems in the class APX that do not admit a PTAS unless $NP = RP$, including a number of interesting special cases of the geometric set cover problem [5], this correspondingly implies that an APX-hard problem cannot admit a PTAS unless $P = NP$.

3 Hardness results

Proposition 1 *For a subgraph G of a $\mathbb{Z}^2$ integer lattice with vertex set V_G, we have that $LPCP(G, \mathcal{L}, \Omega)$ is NP-hard under both the constraint that $|\mathcal{L}| = 1$ and the constraint that $\mathcal{L} = \{1, 2, \ldots, |V_G|\}$.*

Proof. Letting G be a subgraph of a $\mathbb{Z}^2$ integer lattice with vertex set V_G, by the proof argument for "Theorem 7" of Liśkiewicz et. al. [20] we have that there is an efficient polynomial time counting reduction (more specifically, a polynomial time many-one counting "weakly parsimonious" reduction) from counting (case 1) SAWs of a specific length $l_r \in \mathbb{N}$ in G, and (case 2) SAWs of all possible lengths in G, to counting st-Hamiltonian paths in a subcubic planar graph H. The aforementioned proof argument also gives specific polynomial-time computable formula for the number of SAWs that must exist in (case 1) and (case 2), which we will denote T_1 and T_2, respectively, for there to exist at least one st-Hamiltonian path in H. We can also observe, as detailed in "Section 3" of Liśkiewicz et. al. [20], that H is constructed via a polynomial time many-one counting reduction from an instance of $\#3SAT$ to the problem of counting st-Hamiltonian paths in a subcubic planar graph.

Now, let G_1 and G_2 correspond to subgraphs of a $\mathbb{Z}^2$ integer lattice constructed from a subcubic planar graph H for (case 1) and (case 2), respectively, in the proof argument for "Theorem 7" of Liśkiewicz et. al. [20]. Observe that by specifying parameters $\mathcal{L} = \{l_r\}$ and $\Omega = k_B \cdot \ln(T_1 + 1)$, a witness for $LPCP(G_1, \mathcal{L}, \Omega)$ will be the null set if and only if H possesses an st-Hamiltonian path. Similarly, observe that by specifying parameters $\mathcal{L} = \{1, 2, \ldots, |V_G|\}$ and $\Omega = k_B \cdot \ln(T_2 + 1)$, a witness for $LPCP(G_2, \mathcal{L}, \Omega)$ will be the null set if and only if H possesses an st-Hamiltonian path. As the st-Hamiltonian path problem is NP-complete for arbitrary instances of the graph H due to the manner in which the graph is constructed, this yields the proposition. □

Proposition 2 *$LPCP(G, \mathcal{L}, \Omega)$ is NP-hard and APX-hard $\forall \Omega \geq 0$ and for all finite $\mathcal{L} \subset \mathbb{N}_{\geq 2}$.*

Proof. By metatheorems of Yannakakis & Lewis [28, 19] and Lund & Yannakakis [21], we have that the problem of deleting a minimum set of vertices in a simple undirected graph G to satisfy a property Π is NP-hard and APX-hard, respectively, if Π is a *nontrivial* and *hereditary* property. Here, a property being nontrivial means that it both holds and fails to hold for infinitely many graphs, and a property being *hereditary* means that it is satisfied for a graph if and only if it is satisfied for all of the graph's induced subgraphs.

Now, letting $\mathcal{L} \subset \mathbb{N}_{\geq 2}$ be some finite set of vertex-wise lengths for SAWs, observe that there are infinitely many independent sets having no embeddings of SAWs of length $l_i \in \mathcal{L}$, and infinitely many finite undirected graphs having more than an arbitrary number of embeddings of SAWs of length $l_i \in \mathcal{L}$. Accordingly, in the context of the LPCP problem where we model lattice polymers as SAWs having vertex-wise lengths from a set $\mathcal{L}$, $\forall \Omega \geq 0$ we have that there are infinitely many graphs failing to satisfy and satisfying a property Π that $\mathcal{S}_{(G,\mathcal{L})} \leq \Omega$. This implies that the aforementioned property Π is *nontrivial*. We also trivially have that Π is hereditary, as deleting vertices in a graph will cause the number of embeddings of SAWs with lengths in $\mathcal{L}$ to weakly monotonically decrease.

Putting everything together, and recalling that a witness for an instance of $LPCP(G, \mathcal{L}, \Omega)$ is a minimum set of vertices in a simple undirected graph G whose deletion yields a graph G' satisfying the property $\mathcal{S}_{(G',\mathcal{L})} \leq \Omega$, we have that $LPCP(G, \mathcal{L}, \Omega)$ is NP-hard and APX-hard $\forall \Omega \geq 0$ and for all finite $\mathcal{L} \subset \mathbb{N}_{\geq 2}$. □

4 Approximation algorithms for LPCP

Theorem 3 *Letting G is a simple undirected graph with vertex set V_G, edge set E_G, and treewidth $tr(G)$, letting $\zeta_{tw} = f(tr(G)) \cdot \mathcal{O}(|V_G| + |E_G|)$ for some computable function f, and letting $\mathcal{Y}_{init}$ be an initial system configuration entropy, we have that $LPCP(G, \mathcal{L}, \Omega)$ admits an $\mathcal{O}\left(\zeta_{tw} \cdot |V_G|^3 \cdot \ln(|V_G|)\right)$ time $\left(\ln\left(e^{(\mathcal{Y}_{init})} - e^{\Omega}\right)\right)$-approximation algorithm.*

Proof. Interpreting SAWs in G with lengths drawn from the set $\mathcal{L}$ as a universe of elements, and treating each vertex in V_G as the set of SAWs it is covered by, observe that we can correspondingly interpret $LPCP(G, \mathcal{L}, \Omega)$ as a partial set cover problem wherein the objective is to cover at least $\approx \left(e^{(\mathcal{Y}_{init})} - e^{\Omega}\right)$ such elements (i.e., SAW embeddings) with the minimum possible number of sets (i.e., vertices). Accordingly, we immediately have a $\left(\ln\left(e^{(\mathcal{Y}_{init})} - e^{\Omega}\right)\right)$-approximation algorithm as a consequence of the harmonic approximation guarantee for the greedy algorithm for partial set cover [15] (see also Slavík [25] for a detailed performance analysis of the greedy algorithm for the original set cover problem).

In the current context, we can observe that: (obs. 1) there will be an $\mathcal{O}\left(|V_G|^2\right)$ overhead for the subroutines of the greedy algorithm, where for at most $|V_G|$ iterations, we scan at most $|V_G|$ vertices to find the ones whose deletion will maximize coverage of the elements corresponding to SAW embeddings in G; (obs. 2) the selected vertex for each iteration will necessarily be a vertex $v_i \in V_G$ having the largest SAW centrality, $\mathcal{C}_{SAW}(G, \mathcal{L}, v_i)$ (as defined in the introduction of the current work); and (obs. 3) that there will be at most $\mathcal{O}(|V_G|!)$ SAWs of all possible lengths in G, implying that there will be at most the same number of elements to cover in the partial set covering formulation of $LPCP(G, \mathcal{L}, \Omega) \implies$ we will need to read at most the first $\mathcal{O}(|V_G| \cdot \ln(|V_G|))$ bits of each vertex SAW centrality $\mathcal{C}_{SAW}(G, \mathcal{L}, v_i)$ to determine the largest values. Letting Ψ be the cost of computing the SAW centrality for a vertex $v_i \in V_G$, (obs. 1) through (obs. 3) imply that the aforementioned approximation algorithm will have a time complexity of $\mathcal{O}\left(\Phi \cdot |V_G|^3 \cdot \ln(|V_G|)\right)$.

We can now observe the following lemma concerning the treewidth fixed-parameter tractability of computing $\mathcal{C}_{SAW}(G, \mathcal{L}, v_i)$:

Lemma 4 *For a simple undirected graph G with vertex set V_G and edge set E_G, the problem of determining the SAW centrality values for a vertex $v_i \in V_G$, $\mathcal{C}_{SAW}(G, \mathcal{L}, v_i)$, is treewidth FPT, and can be calculated in $\zeta_{tw} = \mathcal{O}(|V_G| + |E_G|)$ time if G has bounded treewidth.*

Proof. It suffices to show there exists a linear time treewidth FPT algorithm for counting the number of SAWs between an arbitrary pair of vertices v_s and v_t in a graph. Observe that we can simply run this procedure for an instance of a graph with or without a specified vertex to determine $\mathcal{C}_{SAW}(G, \mathcal{L}, v_i)$.

We proceed by appealing to an extension of Courcelle's well-known algorithmic metatheorem [6, 7, 8, 9] to counting and optimization problems [1, 10]. In particular, we appeal to "Theorem 32" of Courcelle, Makowsky, & Rotics [9], which states in part that if we can express the existence of a graph property ϕ in the fragment of second order logic denoted "extended" Monodic Second Order (MS_2) (see, e.g., Downey & Fellows [13] for an elaboration), then we are guaranteed an algorithm for this problem having time complexity $c \cdot \mathcal{O}(|V| + |E|)$, where c is a constant that depends only on ϕ and the graph treewidth $tw(G)$. Here, this time complexity is a consequence of the proof being based on the bottom-up traversal of a tree decomposition for a finite simple undirected graph G, which has time complexity linear in the size of the tree, and the existence of an $\mathcal{O}(|V| + |E|)$ algorithm due to Bodlaender [3] for computing a tree decomposition of G having width at most $tw(G)$.

To establish the lemma at hand, it now suffices to note that the existence of a path between an arbitrary pair of vertices v_s and v_t in a graph is expressible in first-order (FO_1) logic. In particular, we refer the reader to "pg. 4" of [7], where Courcelle discusses the use of an FO_1 auxiliary predicate " QuasiPath' " for expressing reachability between a pair of vertices in an undirected graph. □

Putting everything together, we can set $\Phi = \zeta_{tw}$ in the earlier asymptotic time analysis of the $\left(\ln\left(e^{(\mathcal{Y}_{init})} - e^{\Omega}\right)\right)$-approximation algorithm for $LPCP(G, \mathcal{L}, \Omega)$ to yield the time complexity in the statement of the current theorem. □

Theorem 5 *Letting G be a simple undirected graph with vertex set V_G and letting $\mathcal{Y}_{init}$ be an initial system configuration entropy, if an $\mathcal{O}(\Psi)$ deterministic algorithm exists for computing the SAW centrality of a vertex $v_i \in V_G$, $\mathcal{C}_{SAW}(G, \mathcal{L}, v_i)$ with multiplicative error $1 \pm \epsilon$, then we correspondingly have that $LPCP(G, \mathcal{L}, \Omega)$ admits an $\mathcal{O}\left(\Psi \cdot |V_G|^3 \cdot \ln(|V_G|)\right)$ time $\left(\frac{\ln\left(e^{(\mathcal{Y}_{init})} - e^{\Omega}\right)}{1-2\epsilon}\right)$-approximation algorithm.*

Proof. Recalling our earlier reformation of $LPCP(G, \mathcal{L}, \Omega)$ as a partial set cover problem in the proof argument for Theorem 3, we begin by observing the following lemma:

Lemma 6 *Letting $\mathcal{P}$ be an instance of the partial set cover problem, where $\mathcal{U}$ is the universe of elements, X is a collection of sets of elements from $\mathcal{U}$, and $0 \leq p \leq 1$*

is the fraction of elements that must be covered, and letting $f_{\epsilon-greedy}$ be an instance of the greedy algorithm, which in each iteration selects a set uniformly at random from all sets in X covering a fraction $(1-2\epsilon)$ of the maximum possible number of elements that can covered in the iteration, we have that $f_{\epsilon-greedy}$ will be a $\left(\frac{\ln(p)}{2\epsilon-1}\right)$-approximation algorithm for $\mathcal{P}$.

Proof. Letting α be the size of the minimum partial set cover for $\mathcal{P}$, observe that the kth iteration of $f_{\epsilon-greedy}$ will, in the worst case, reduce the number of uncovered elements in $\mathcal{U}$ by a fraction $\left(1-\frac{1-2\epsilon}{\alpha}\right)$. Accordingly, we can express the number of uncovered elements in $\mathcal{U}$ after r iterations of $f_{\epsilon-greedy}$ as $|\mathcal{U}|\cdot\left(1-\frac{1-2\epsilon}{\alpha}\right)^r$, or equivalently, as $|\mathcal{U}|\cdot\left(\left(1-\frac{1-2\epsilon}{\alpha}\right)^\alpha\right)^{\frac{r}{\alpha}}$.

We next establish that $\left(1-\frac{1-2\epsilon}{\alpha}\right)^\alpha$ will weakly monotonically increase with α for $0\leq\epsilon\leq 1$ and $\alpha\geq 1$. To begin, we can note that:

$$\frac{\partial}{\partial\alpha}\left[\left(1-\frac{1-2\epsilon}{\alpha}\right)^\alpha\right]\geq 0$$
$$\Longleftarrow \left(\frac{1}{\alpha}\right)\cdot\left(\frac{\alpha+2\epsilon-1}{\alpha}\right)^{(\alpha-1)}\cdot$$
$$\left((1-2\epsilon)+(\alpha+2\epsilon-1)\cdot\ln\left(\frac{\alpha+2\epsilon-1}{\alpha}\right)\right)\geq 0$$
$$\Longleftarrow \left((1-2\epsilon)+(\alpha+2\epsilon-1)\cdot\ln\left(\frac{\alpha+2\epsilon-1}{\alpha}\right)\right)\geq 0$$

Now let $\omega=\left((1-2\epsilon)+(\alpha+2\epsilon-1)\cdot\ln\left(\frac{\alpha+2\epsilon-1}{\alpha}\right)\right)$. Here, we can observe that $\frac{\partial}{\partial\epsilon}(\omega)=2\ln\left(\frac{\alpha+2\epsilon-1}{\alpha}\right)$, and accordingly, that for fixed $\alpha\geq 1$, the expression ω will be minimized for $\epsilon=\frac{1}{2}$. As $\epsilon=\frac{1}{2}\implies\omega=0$, we therefore have that ω is non-negative whenever $\alpha\geq 1$ and $0\leq\epsilon\leq 1$, and therefore that $\frac{\partial}{\partial\alpha}\left[\left(1-\frac{1-2\epsilon}{\alpha}\right)^\alpha\right]$ will be non-negative $\forall\alpha\geq 1$. It now suffices to note that $\alpha=1$ and $0\leq\epsilon\leq 1\implies\left(1-\frac{1-2\epsilon}{\alpha}\right)^\alpha\geq 0$.

Putting everything together, we can use the approximation $\lim_{\alpha\to\infty}\left(1-\frac{1-2\epsilon}{\alpha}\right)^\alpha=e^{(2\epsilon-1)}$ to express the fraction of covered elements after r iterations of $f_{\epsilon-greedy}$ as $|\mathcal{U}|\cdot\left(e^{(2\epsilon-1)}\right)^{\frac{r}{\alpha}}$. Thus, $|\mathcal{U}|\cdot\left(e^{(2\epsilon-1)}\right)^{\frac{r}{\alpha}}=p\cdot|\mathcal{U}|\implies r=\left(\frac{\alpha\cdot\ln(p)}{2\epsilon-1}\right)$, yielding the lemma. □

To establish the theorem at hand, following the proof argument for Theorem 3, it now suffices to observe that $p\cdot|\mathcal{U}|$ from Lemma 6 can be understood to correspond to $\left(e^{(\mathcal{Y}_{init})}-e^{\Omega}\right)$, and that $f_{\epsilon-greedy}$ from Lemma 6 can be understood to correspond to the $\mathcal{O}(\Psi)$ deterministic algorithm for computing $\mathcal{C}_{SAW}(G,\mathcal{L},v_i)$ with multiplicative error $1\pm\epsilon$. □

Corollary 7 *There exists an instance of the $\mathcal{O}\left(\Psi\cdot|V_G|^3\cdot\ln(|V_G|)\right)$ time $\left(\frac{\ln\left(e^{(\mathcal{Y}_{init})}-e^{\Omega}\right)}{1-2\epsilon}\right)$-approximation algorithm for $LPCP(G,\mathcal{L},\Omega)$ from Theorem 5, where letting G be a simple undirected graph with vertex set V_G and edge set E_G, we have that $\Psi\in\mathcal{O}\left(\sum_{i=1}^{|\mathcal{L}|}\left(4^{l_i+\mathcal{O}\left(\sqrt{l_i}\cdot\left(\ln^2(l_i)+\ln^2\left(\frac{1}{\epsilon}\right)\right)\right)}\right)\cdot|E_G|\cdot\ln(|V_G|)\right)$.*

Proof. This result follows directly from a recent result of Björklund et. al. [2] that an $\mathcal{O}\left(\left(4^{k+\mathcal{O}\left(\sqrt{k}\cdot\left(\ln^2(k)+\ln^2\left(\frac{1}{\epsilon}\right)\right)\right)}\right)\cdot|E_G|\cdot\ln(|V_G|)\right)$ time and polynomial-space deterministic PTAS exists for counting the number of length k SAWs in a simple undirected graph G with vertex set V_G and edge set E_G. □

Corollary 8 *For a variant of $LPCP(G,\mathcal{L},\Omega)$ where we delete edges in lieu of vertices, the time complexities of the Theorem 3 and Theorem 5 approximation algorithms become $\mathcal{O}\left(\zeta_{tw}\cdot|E_G|\cdot|V_G|^2\cdot\ln(|V_G|)\right)$ and $\mathcal{O}\left(\Psi\cdot|E_G|\cdot|V_G|^2\cdot\ln(|V_G|)\right)$, respectively.*

Proof. Observe that we can measure the SAW centrality of edges in a lattice or graph G in exactly the same manner (and with the same time complexity) as we computed the SAW centrality of vertices – e.g., by simply computing the change in the number of relevant SAW embeddings with and without a given edge being present. Therefore, the only change in the time complexity for the Theorem 3 and Theorem 5 approximation algorithms comes from having to compute the SAW centralities of $|E_G|$ edges instead of $|V_G|$ vertices. □

Remark 1 *For a variant of $LPCP(G,\mathcal{L},\Omega)$ where we consider the configuration entropies of "rigid" lattice polymers (e.g., lattice proteins) where every embedding must satisfy a set of consecutive dihedral angles for bond edges, for an interpretation of "rigid" lattice polymers as SAWs required to have a specific geometry when embedded in a lattice or graph, the time complexities of the Theorem 3 and Theorem 5 approximation algorithms becomes $\mathcal{O}\left(|E_G|\cdot|V_G|^3\cdot\ln(|V_G|)\right)$ and $\mathcal{O}\left(|E_G|\cdot|V_G|^3\cdot\ln(|V_G|)\right)$, respectively.*

Proof. It suffices to observe that if we require SAWs to have a specific geometry, we can trivially enumerate the number of embeddings of such SAWs in $\mathcal{O}(|E_G|)$ time, as any edge of a specific SAW will fix the remaining edges. The stated changes in the time complexities for the Theorem 3 and Theorem 5 approximation algorithms then follow as a consequence of removing the cost of computing SAW centralities. □

5 Concluding Remarks

For a universe of elements $\mathcal{U}$, the general set cover problem is known not to be approximable within a factor of $(1-o(1))\cdot\ln(|\mathcal{U}|)$ unless $P=NP$ [12]. Accordingly, as

we establish the Theorem 3 and Theorem 5 approximation algorithms via reduction to equivalent partial set cover problems, it is unlikely that we can significantly improve the current approximation guarantees in either case. However, concerning a future research direction, we remark that much better performance guarantees can be achieved for the geometric set cover problem (see, e.g., Brönnimann & Goodrich [4]). Here, it should be possible to take advantage of a particular embedding of a lattice or graph to treat sets of vertices or SAWs (e.g., in a geometric hitting set formulation) as polygons or other shapes, and in some cases achieve better approximation guarantees or time complexities.

References

[1] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991.

[2] A. Björklund, D. Lokshtanov, S. Saurabh, and M. Zehavi. Approximate counting of k-paths: simpler, deterministic, and in polynomial space. *ACM Trans. Algorithms*, 17(3):26:1–26:44, 2021.

[3] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[4] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite VC-dimension. *Discrete Comput. Geom.*, 14(1):463–479, 1995.

[5] T. M. Chan and E. Grant. Exact algorithms and APX-hardness results for geometric packing and covering problems. *Comput. Geom.*, 47(2):112–124, 2014.

[6] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Inform. Comput.*, 85(1):12–75, 1990.

[7] B. Courcelle. On the expression of graph properties in some fragments of monadic second-order logic. *In: (N. Immerman and P. G. Kolaitis, eds.) Descriptive Complexity and Finite Models, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 31, 1997.

[8] B. Courcelle. Graph structure and monadic second-order logic: language theoretical aspects. *Proc. of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1–13, 2008.

[9] B. Courcelle, J. A. Makowsky, and U. Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete Appl. Math.*, 108(1-2):23–52, 2001.

[10] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms, 1st edition.* Springer: Switzerland, 2015.

[11] R. Diestel. *Graph theory, 5th edition.* Springer-Verlag: Heidelberg, 2017.

[12] I. Dinur and D. Steurer. Analytical approach to parallel repetition. *Proc. 46th annual ACM Symposium on Theory of Computing (STOC)*, pages 624–633, 2014.

[13] R. G. Downey and M. R. Fellows. *Fundamentals of parameterized complexity, 1st edition.* Springer: New York, NY, 2013.

[14] S. P. N. Dubey, N. G. Kini, S. Balaji, and M. S. Kumar. A review of protein structure prediction using lattice model. *Crit. Rev. Biomed. Eng.*, 46(2):147–162, 2018.

[15] T. Elomaa and J. Kujala. Covering analysis of the greedy algorithm for partial cover. *Algorithms and Applications. Lecture Notes in Computer Science (LNCS)*, 6060:102–113, 2010.

[16] P. J. Flory. *Principles of polymer chemistry (1st edition).* Cornell University Press: Ithaca, NY, 1953.

[17] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[18] A. Landherr, B. Friedl, and J. Heidemann. A critical review of centrality measures in social networks. *Bus. Inf. Syst. Eng.*, 2:371–385, 2010.

[19] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980.

[20] M. Liśkiewicz, M. Ogihara, and S. Toda. The complexity of counting self-avoiding walks in subgraphs of two-dimensional grids and hypercubes. *Theoret. Comput. Sci.*, 304(1-3):129–156, 2003.

[21] C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. *Proc. 20th International Colloquium on Automata, Languages, and Programming (ICALP)*, 700:40–51, 1993.

[22] N. Madras and G. Slade. *The self-avoiding walk, 1st edition.* Birkhäuser: Boston, MA, 1996.

[23] M. Mann, R. Saunders, C. Smith, R. Backofen, and C. M. Deane. Producing high-accuracy lattice models from protein atomic co-ordinates including side chains. *Adv. Bioinform.*, 2012(Article ID 148045):1–6, 2012.

[24] M. Mann, C. Smith, M. Rabbath, M. Edwards, S. Will, and R. Backofen. CPSP-web-tools: a server for 3D lattice protein studies. *Bioinformatics*, 25(5):676–677, 2009.

[25] P. Slavík. A tight analysis of the greedy algorithm for set cover. *J. Algorithms*, 25(2):237–254, 1997.

[26] S. W. P. Turner, M. Cabodi, and H. G. Craighead. Confinement-induced entropic recoil of single DNA molecules in a nanofluidic structure. *Phys. Rev. Lett.*, 88(12):128103:1–128103:4, 2002.

[27] D. R. White and S. P. Borgatti. Betweenness centrality measures for directed graphs. *Social Networks*, 16(4):335–346, 1994.

[28] M. Yannakakis. Node- and edge-deletion NP-complete problems. *Proc. 10th annual ACM Symposium on Theory of Computing (STOC)*, pages 253–264, 1978.