USING THE COULLARD-PULLEYBLANK RELAXED CYCLE CONE TO PROVE THE INTRACTIBILITY OF UNIFORMLY SAMPLING FROM THE VERTICES OF A POLYTOPE

LORENZO NAJT

Abstract. This is a cleaned up version of an argument posted on [3]. These notes are a companion document to "The intractability of uniformly sampling and counting the vertices of a polytope," which proves stronger results with less technical intricacies.

Definition 0.1. *If $S \subseteq T$ is a subset of set, then χ_S denotes the indicator function of S inside the vector space $\mathbb{R}^T$.*

0.1. Review of convex geometry.

In this section we record some basic definitions and facts about the geometry of polytopes. The main thing we need from this section for the proof is Proposition 0.16.

Definition 0.2 (Cone). *A subset $C \subseteq \mathbb{R}^n$ is a cone if $\forall v, w \in C$ and $\lambda \geq 0$, $v + w \in C$ and $\lambda v \in C$. The dimension of a cone is the vector space dimension of its linear span. A cone is* Strongly convex *if it contains no linear subspace.*

Definition 0.3 (Ray). *A ray in $\mathbb{R}^n$ is a set of the form $\{\lambda v : \lambda \geq 0\}$, where $v \neq 0$ is some fixed vector. Given a vector v, let $[v]$ denote the ray spanned by v, $[v] = \{\lambda v : \lambda \geq 0\}$. In particular, a ray is a strongly convex one-dimensional cone.*

Definition 0.4 (Faces). *Let C be a cone in $V = \mathbb{R}^n$. The faces of C are all the sets of the form $u^T \cap C$, where $c \in V^*$ is a vector with $\langle u, v \rangle \geq 0$ for all $v \in V$.*

Definition 0.5 (Conical hull). *Let $W = \{w_1, \dots, w_m\} \subseteq \mathbb{R}^n$ be a set of vectors. Let $Cone(W) = \{\sum_{i=1}^m \lambda_i w_i : \lambda_i \geq 0\}$, which is the cone generated by W.*

Definition 0.6 (Conically independent). *Let $W = \{w_1, \dots, w_m\} \subseteq \mathbb{R}^n$ be a set of vectors. If for all i, $Cone(W \setminus \{w_i\}) \not\ni w_i$, then we say that W is conically independent.*

Definition 0.7 (Extremal ray). *Let C be a strongly convex cone. The extremal rays are the one dimensional faces of C. We denote them by $\text{Ext}(C)$.*

It will be convenient to define the extremal rays of a cone as defined by a set of conically independent, generating sets for C:

Lemma 0.8. *Suppose that $C = Cone(F)$, where F is a set of conically independent vectors. Then $\{\{\lambda v : \lambda \geq 0\} v \in F\}$ is the set of extremal rays of $Cone(F)$.*

Definition 0.9 ($\mathcal{H}$-polyhedron / polytope). *A $\mathcal{H}$-polyhedron is the intersection of a finite set of half-spaces. It is called a polytope if it is bounded. The dimension is the dimension of its affine span.*

Definition 0.10 (Faces of a polytope). *If P is a polytope, and u is an affine linear functional which is non-negative on P, then $\{u = 0\} \cap P$ is a face of P.*

Definition 0.11 (Vertices of a polytope). *Let P be a polytope. A vertex is a zero dimensional face of P. Let $\|(P)$ denote the set of vertices of P.*

Definition 0.12 (Convex hull, Convex independence). *Let $X \subseteq \mathbb{R}^n$. The convex hull , $ConvHull(X)$ is the smallest convex set containing X. X is said to be convexly independent if for all $x \in X$, $x \notin ConvHull(X \setminus x)$.*

It will be convenient to characterize the vertices of a polytope P as the set of convexely independent points which have P in their convex hull.

Definition 0.13 (Dual Cone/ Polytope). *Let C be a cone in a vector space V. Then $C^\vee = \{u \in V^* : u(c) \geq 0, \forall c \in C\}$ is the dual cone of C. It is also a convex cone.*

Definition 0.14 (Vertex figure). *Let C be a cone. Let $u \in C^\vee$ and which has $u^T \cap C = \{0\}$. We define $C_u = C \cap \{x \in \mathbb{R}^n : u(x) = 1\}$, and call it the vertex figure defined by u.*

The choice of u will not affect the combinatorics in any way that matters to our reduction (Proposition 0.16). In particular, all vertex figures are projectively equivalent. Hence we will refer to the projective isomorphism class of any vertex figure as *the* vertex figure of C. To begin with, the vertex figure is always a polytope:

Lemma 0.15 (Vertex figure is a polytope). *Suppose that $f \in C^\vee$,with $C = Cone(\{v_1, \dots, v_m\})$ and $\{f = 0\} \cap C = \{0\}$ then $C_f = ConvHull(\{\frac{v_i}{f(v_i)} : i = 1, \dots, m\}$. In particular, C_f is a polytope.*

Proof. First we observe that $f(v_i) > 0$ for each i, as otherwise if $f(v_i) = 0$ for some i, then $\{f = 0\} \cap C$ contains a ray. Now, let $v \in C \cap \{f = 1\}$, then $v = \sum \lambda_i \frac{v_i}{f(v_i)}$ for some $\lambda_i \geq 0$. Applying f to both sides of this equation shows that $\sum \lambda_i = 1$, hence $v \in ConvHull(\{\frac{v_i}{f(v_i)} : i = 1, \dots, m\}$. The converse is clear as $\{f = 1\} \cap C$ is a convex set and contains each $\frac{v_i}{f(v_i)}$. □

Next, we argue that the calculation in the previous lemma actually gives a bijection between the extremal rays and the vertices of the vertex figure.

Proposition 0.16 (Vertices of the vertex figure, and extremal rays). *Let C be a strongly convex cone, and C_f a vertex figure. Then the map $\|(C_f) \to Ext(C)$ given by sending a vector v to the ray $\{\lambda v : \lambda \geq 0\}$ is a bijection, with inverse sending the ray generated by v to $\frac{v}{f(v)}$.*

Proof. Let F be a generating set of $\tilde{C}(G)$. We have already shown that $C_f = Hull(\{\frac{v}{f(v)} : v \in F\}$. Let $X = \{\frac{v}{f(v)} : v \in F\}$. To finish the argument, it would be sufficient to argue that X is convexly independent, since the vertices of a polytope P are the only convexly independent convex generating set of P. The convex independence of X, however, follows immediately from the conical independence of F. Thus $X = Vert(C_f)$.□

0.2. Generalities on Sampling. In this section, we discuss some background on sampling problems, the class RP, why RP $\neq$ NP is a reasonable assumption, and what it means for a sampling problem to be intractable. We also prove lemmas that will be used throughout.

The formalism for sampling problems, which goes back to at least [5], begins with a finite alphabet Σ and a binary relation between words in this alphabet $R \subseteq \Sigma^* \times \Sigma^*$. We interpret $(x, y) \in R$ as asserting that y is a solution to the instance x. For example, we can define a binary relation R as those (x, y) such that x encodes a graph $G(x)$ and y encodes the edges of a simple cycle of $G(x)$. We will consider only those relations that can be verified efficiently, which are called p-relations:

Definition 0.17 (p-relations, [5]). *A relation $R \subseteq \Sigma^* \times \Sigma^*$ is a p-relation if there is a deterministic polynomial time Turing machine that recognizes $R \subseteq \Sigma^* \times \Sigma^*$ and if there is a polynomial p such that $\forall x$, $(x, y) \in R$ implies that $|y| \leq p(|x|)$. We define $R(x) = \{y \in \Sigma^* : (x, y) \in R\}$.*

Now we define the sampling problems we will be considering:

Definition 0.18 (Family of p-distributions). *A family of p-distributions is defined by a p-relation R and function $f : \Sigma^* \to \mathbb{Q}_{\geq 0}$. For each instance $x \in \Sigma^*$ with $R(x) \neq \emptyset$, we require that f is not identically zero on $R(x)$. For such an instance x, we associate a probability distribution p_x on $R(x)$, where $y \in R(x)$ has weight proportional to $f(y)$. The uniform distribution on R is defined by taking f to be identically* 1.

Definition 0.19 (Sampling problem). *To each family of p-distributions (R, p_x), there is an associated sampling problem, which we also refer to as (R, p_x):*

$P = (R, p_x)$ Sampling

Input: $x \in \{x \in \Sigma^* : R(x) \neq \emptyset\}$

Output: A sample drawn according to p_x.

Similar to approximation algorithms in the deterministic case, we can ask if Turing machine "almost" solves a sampling problem:

Definition 0.20 (α-almost solving a sampling problem). *Suppose that $P = (R, p_X)$ is some sampling problem. Let $\alpha \in [0, 1]$. We say that a probabilistic Turing machine M α-*almost solves *$P = (R, p_x)$* Sampling *if for all instances X with $R(X) \neq \emptyset$, $M(X)$ accepts X at least half the time and then outputs a sample from a distribution q_X^M, where $||q_X^M - p_X||_{TV} \leq \alpha$. In the case $\alpha = 0$, we say that M solves the sampling problem.*

We will use the complexity class RP to describe the intractability of a sampling problem.

Definition 0.21 (The class RP [2]). *RP is the class of languages $L \subseteq \Sigma^*$ such that there is a polynomial time probabilistic Turing machine M and a constant $\epsilon > 0$ so that, if $x \notin L$, $M(x)$ always rejects, and if $x \in L$, $M(x)$ accepts with probability at least ϵ.*

It is widely believed that RP $\neq$ NP; this belief follows from the widely believed conjectures that NP $\neq$ P [1] and BPP $=$ RP $=$ P [4]. To show that a sampling problem is likely to be intractable, it is common ([5, Proposition 5.1] or [9, Theorem 1.17]) to show that the existence of an efficient sampler would imply RP $=$ NP. This is how we proceed.

0.3. The Coullard-Pulleyblank relaxed cycle cone. Let $G = (V, E)$ be a finite graph.

Definition 0.22 (Simple cycles; Cycle Cone). *A simple cycle is a set of edges of G which induce a cycle graph. Let $\mathrm{Cycle}(G)$ denote the cone in $\mathbb{R}^E$ generated by the indicator functions of all simple cycles of G. $\mathrm{Cycle}(G)$ is called the cycle cone of G.*

A key gadget used in our reduction is derived from a relaxation of the cycle cone introduced by Coullard and Pulleyblank in [7]. This relaxation is obtained by dropping all but a comparatively small set of equations from equations defining the cycle cone. First, we recall a well known description of the facets of the cycle cone.

Theorem 0.23 (Seymour's equations for the cycle cone [8]; [7]). *Let $\mathscr{K}$ be the set of edge cuts in G; that is, sets of the form $\mathrm{cut}(U, U^c)$ for any set of vertices U. Then, $\mathrm{Cycle}(G)$ is defined by the following system of inequalities:*

$$\begin{cases} x(K \setminus e) - x(e) \geq 0 \textit{ for all } K \in \mathscr{K} \textit{ and all } e \in K \\ x(e) \geq 0 \textit{ for all } e \in E \end{cases} \tag{0.1}$$

The relaxation of $\mathrm{Cycle}(G)$ we will use is obtained by only using the cut sets defined by single vertices. For a vertex v, let $\delta(v) = \mathrm{cut}(v, V \setminus v)$. In [7] Coullard and Polleyblank defined the following relaxation of Seymour's system:

$$\begin{cases} x(\delta(v) \setminus e) - x(e) \geq 0 \text{ for all } v \in V \text{ and all } e \in \delta(v) \\ x(e) \geq 0 \text{ for all } e \in E \end{cases} \tag{0.2}$$

Definition 0.24 (Relaxed cycle cone). *Let $\tilde{C}(G)$ denote the set of solutions to 0.2.*

Coullard and Pulleyblank characterized the extremal rays of the relaxed cycle cone in the following way:

Theorem 0.25 ([7]). *The cone $\tilde{C}(G)$ is generated by the set $\{\chi(C) : C \subset \mathscr{C}\}$, together with the set of vectors of the form $\chi(C_1) + \chi(C_2) + 2\chi(P)$, where C_1 and C_2 are node-disjoint cycles and P is a path joining C_1 and C_2 and having no internal nodes in C_1 or C_2.*

The polytope in our reduction will be obtained from the relaxed cycle cone as a vertex figure. In particular, as we will review below, Theorem 0.25 implies that the vertices of the vertex figure correspond bijectively to certain combinatorial objects associated to G. Thus, sampling from that combinatorial set can be reduced to the vertex sampling problem. Once that connection is established, we will use standard techniques [5,6] to concentrate the probability of the sampler onto the maximizers of an NP-hard optimization problem. Since $\tilde{C}(G)$ is in the non-negative orthant, the following polytope describe a vertex figure of $\tilde{C}(G)$ (subsection 0.1).

Definition 0.26 (Relaxed Cycle Polytope). *Let $G = (V, E)$ be a graph. We define $P(G) = \tilde{C}(G) \cap \{x : \sum_{e \in E} x(e) = 1\}$.*

Proposition 0.27. *$P(G)$ is a vertex figure of the cone $\tilde{C}(G)$ at 0.*

Proof. Note that because of the constraints $x \geq 0$ on $P(G)$ it follows that the function $f(x) = \sum_{e \in E} x(e)$ is in $\tilde{C}(G)^\vee$, and that $\{f = 0\} \cap \tilde{C}(G) = \{0\}$. Thus, the claim follows from the definition of a vertex figure (Definition 0.14). □

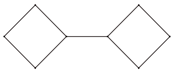


Figure 0.1: A lasso

We note that by introducing slack variables $s_{v,e} = x(\delta(v) \setminus e) - x(e)$ we can express this polytope in equational form. Thus, if we prove intractability results about sampling from the vertices of the polytopes $P(G)$, we will obtain intractability results about `UniformVertex`, defined in terms of equational form polytopes.

0.4. Simple cycles and lassos. To capture the combinatorial description of extremal rays of $\tilde{C}(P)$ given by 0.25, we make the following definitions:

Definition 0.28 (Simple cycle, f_J). *Let $G = (V, E)$ be a graph. A set $J \subseteq E$ is a simple cycle if the subgraph it induces is isomorphic to a cycle graph; that is, if it is connected and each node has degree 2. Given a cycle J, denote $f_J = \chi_J$.*

Definition 0.29 (Lasso, f_L). *Consider edge subgraphs of the form $C_1 \cup P \cup C_2$, where C_1, C_2 are non-empty, vertex disjoint simple cycles and P is a non-empty path connecting them (but is otherwise vertex disjoint from both). We will call such sets of edges* lassos*; see Figure 0.1 for an illustration. For $L = C_1 \cup P \cup C_2$ a lasso, denote $f_L := \chi_{C_1} + 2\chi_P + \chi_{C_2}$.*

Definition 0.30 (SCL). *For a graph W, let $SC(W)$ be the set of non-empty simple cycles, $L(W)$ be the set of lassos, and $SCL(W) = SC(W) \cup L(W)$. We note that $SC(W) \cap L(W) = \emptyset$, since lassos have two nodes of degree 3, so f_J is well defined for any $J \in SCL(W)$ by Definition 0.28 and Definition 0.29.*

We may restate theorem 0.25 in this language:

Theorem 0.31 ([7]). *$\tilde{C}(G)$ is generated as a cone by the set $\{f_J : J \in SCL(W)\}$.*

We also need the following:

Proposition 0.32. *Every vector of the form f_J for $J \in SCL(W)$ generates an extremal ray of $\tilde{C}(W)$.*

Proof. It suffices to show that $\{\chi_C : C \in SC(W)\} \cup \{f_L : L \in L(W)\} = \{f_J : J \in SCL(W)\}$ is a conically independent set. The main idea of the proof is to consider the supports of these vectors, and observe that it is impossible to arrange the supports of a set of simple cycles and lassos so as to perfectly cover up the support of a simple cycle or lasso not represented in that set. We now verify this.

Suppose that for some $D \in SC(G)$,: $\chi_D = \sum_{C \in SC(G)} \alpha_C \chi_C + \sum_{C_1 \cup P \cup C_2 \in Lassos(G)} \beta_{C_1 \cup P \cup C_2}(\chi_{C_1} + 2\chi_P + \chi_{C_2})$ with $\alpha, \beta \geq 0$. By considering the support, we see that for any $C \not\subset D$ we have $\alpha_C = 0$. Thus, the only cycle C with $\alpha > 0$ is D. Moreover, as no lasso is contained in any simple cycle, we have $\beta_L = 0$ for all lassos L.

Now suppose that we have $\chi_{D_1} + 2\chi_Q + \chi_{D_2} = \sum_{C \in SC(G)} \alpha_C \chi_C + \sum_{C_1 \cup P \cup C_2 \in Lassos(G)} \beta_{C_1 \cup P \cup C_2}(\chi_{C_1} + 2\chi_P + \chi_{C_2})$. Again by considering the support, we would be forced to have $C \subset D_1 \cup Q \cup D_2$ for any C with $\alpha_C > 0$. Hence $\alpha_C > 0$ implies that $C \in \{D_1, D_2\}$. Similarly, the only lasso L that can have β_L positive is L. Thus, we would obtain an equation of the form $\chi_{D_1} + 2\chi_Q + \chi_{D_2} = \alpha_1 \chi_{D_1} + \alpha_2 \chi_{D_2} + \beta(\chi_{D_1} + 2\chi_Q + \chi_{D_2})$. By independence of $\chi_{D_1}, \chi_Q, \chi_{D_2}$ (they are all edge disjoint), we conclude that $\beta = 1$, hence $\alpha_1 = \alpha_2 = 0$. □

From the previous two propositions, we obtain the following description of the extremal rays of $\tilde{C}(G)$.

Proposition 0.33. *The function $F : SCL(W) \to \mathrm{Ext}(\tilde{C}(G))$ given by sending J to $[f_J]$ is a bijection. The inverse of F sends $v \in E^{\mathbb{R}}$ to the x_e variables in its support.*

Proof. The previous two propositions tell us that F is surjective. F is injective because a SCL J is determined by the support of f_J. □

Now we articulate the crucial observation, linking the vertices of a polytope to a set of combinatorial gadgets about which we will be able to prove an NP-hardness result in 0.5.

Theorem 0.34. *The vertices of $P(G)$ are in a polynomial time computable bijection with the extremal rays of $\tilde{C}(G)$, which are in a polynomial computable bijection with $SCL(G)$.*

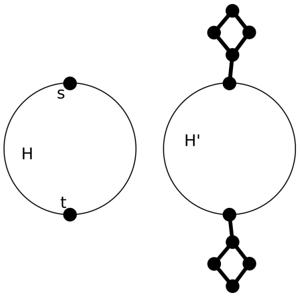


Figure 0.2: Constructing H' from H

Proof. The first bijection follows because $P(G)$ is the vertex figure of $\tilde{C}(G)$, proven in Proposition 0.27, and because the vertices of the vertex figure are in a polynomial time computable bijection with the extremal rays of the corresponding cone. The second bijection follows from Proposition 0.33.

Therefore, to show that it is intractable to sample from the vertices of $P(G)$, it would suffice to show that it is intractable to sample from $SCL(G)$. We now introduce the computational problem we will study in the next section, and formalize the connection to `UniformVertex`.

UniformSCL

Input: A graph G.

Output: An element from the set $SCL(G)$ drawn according to the uniform distribution over $SCL(G)$.

Corollary 0.35. *Suppose that there is a polynomial time probabilistic Turing machine which α almost solves the problem `UniformVertex`. Then there is a polynomial time probabilistic Turing machine which α-almost solves the problem `UniformSCL`.*

0.5. The spanning SCL problem is NP-complete.

Spanning SCL

Input: A graph G

Output: Whether there is an element of $SCL(G)$ which contains an edge adjacent to every node of G.

Proposition 0.36. *SpanningSCL is NP-complete.*

Proof. We make a reduction from Hamiltonian st-path. Let H be a graph. Add a lollipop to s and to t as in Figure 0.2, to get a graph we will call H'. Then H' has a Spanning SCL iff H has a Hamiltonian st-path. □

We will have need to refer to the graphs H' arising in the previous reduction below. Therefore, we make the following definition:

Definition 0.37 (Lollipop ear graphs). *A graph H' obtained from a graph H as in Proposition 0.36 is called a lollipop ear graph. That is, a graph G is a lollipop ear graph if and only if there is a graph H, with vertices s and t in $V(H)$, such that G is isomorphic to the graph obtained by adding lollipops to s and t as in Figure 0.2.*

In fact, we will need a slight strengthening of Proposition 0.36:

Proposition 0.38. *SpanningSCL is* NP*-complete on the class of lollipop ear graphs..*

Next, we will show how to exploit a probability concentration idea as in [5] can be used to show that if we can uniformly sample from $SCL(W)$ for any W, then we can construct an RP algorithm for `SpanningSCL` on lollipop ear graphs. There are some subtle ways that shapes can degenerate, which we work out carefully

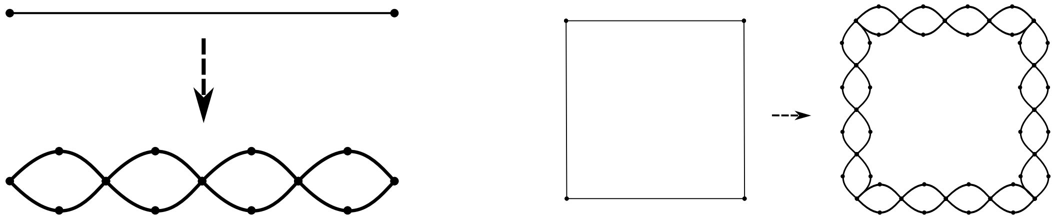
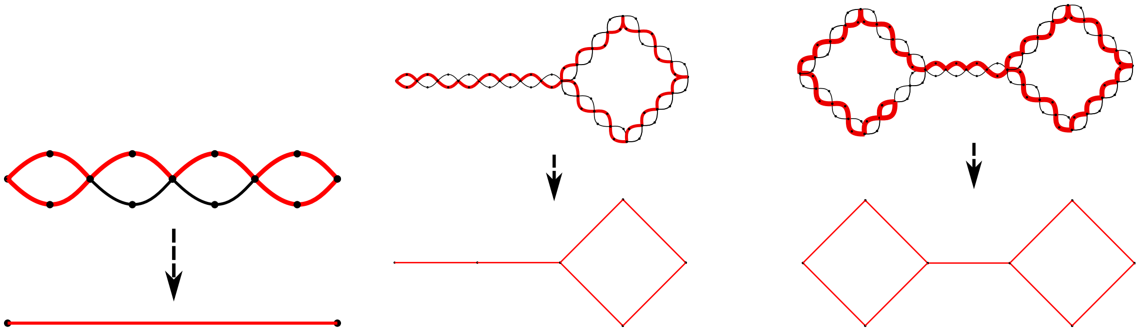


Figure 0.3: Chain of diamonds construction



a) Degenerating to an edge b) Degenerating to a lollipop c) Example of subtle non-degeneration

Figure 0.4: Examples of applying π_4 to lassos

in the next section.

0.6. Probability Concentration on SCL**.** We now build the tools to show that uniformly sampling SCL is intractable.

Definition 0.39 (Chain of Bigons). *Let G be a graph. Let G_d denote the graph obtained from G by replacing each edge with a chain of d diamonds, see (Fig.0.3). For an edge e, let $D_d(e)$ denote the set of edges in the chain of diamonds that replaced it.*

Definition 0.40 (Projection map). *Define a map $\pi_d : SCL(G_d) \to 2^{E(G)}$ by $\pi(X) = \{e \in E(G) : X \cap D_d(e) \neq \emptyset\}$. If clear from context, we call this π.*

We note that $\pi_d(X)$ does not have to be a simple cycle or a lasso. For example, $\pi_d(X)$ could be a single edge, as in Figure 0.4a), or become a lollipop shape, as in Figure 0.4b). However, we note that the image can remain a lasso even if it seems like one of the cycle is collapsed, as in Figure 0.4c). In the next section, will classify the shapes that can occur. Following, we will use this classification to assist our counting.

0.6.1. Classification of the shapes $\pi(X)$. To organize the classification, we shall work with the homeomorphism type of the image $\pi_d(X)$, under the usual association of a graph to a topological space.

Definition 0.41 (Leaves). *Given a graph Y, a vertex in Y is said to be a leaf vertex if it has degree one. Let $L(Y)$ denote the set of leaves of Y, and $l(Y) = |L(Y)|$.*

Definition 0.42 (Cycle rank). *Let G be a graph. The cycle rank is the minimum number of edges one needs to remove from G in order to obtain a tree. It is the same as the rank of the first homology group, hence we denote it by $h_1(G)$.*

Key to classifying the shapes of $\pi(X)$ is the following lemma. We identify $\pi_d(X)$ with the subgraph of G that induces.

Lemma 0.43. *If $X \in SCL(D_d(G))$, then $\pi(X)$ is a connected graph, and $l(\pi(X)) + h_1(\pi(X)) \leq 2$.*

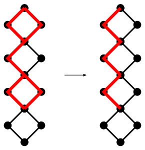


Figure 0.5: Removing halfs

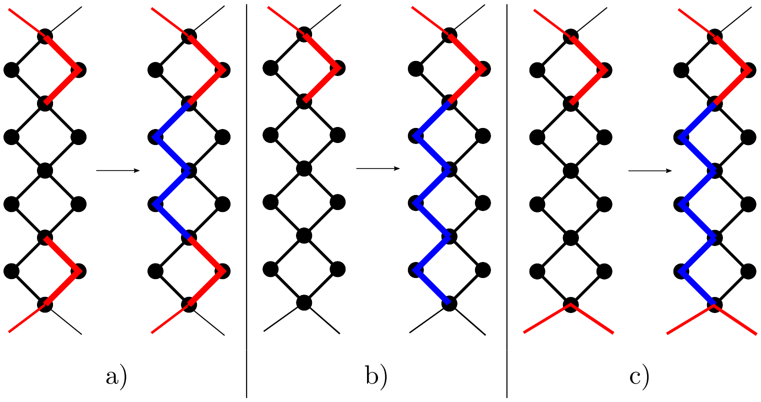


Figure 0.6: Extending to path

Proof. First, $\pi(X)$ is connected, as for any $e_1, e_2 \in E(\pi(X))$ with $e_i' \in D_d(e) \cap X$,a path in X from e_1' to e_2' induces one between e_1 and e_2. If X is a simple cycle, then $\pi(X)$ is either an edge or a simple cycle, and hence satisfies the inequality. To handle the case of a lasso, we will decompose the map π_d into a composition of maps obtained by collapsing the chain of diamonds for each edge at a time. After collapsing each edge, we will verify that the inequality remains true. So, suppose that H' is obtained from H by replacing one edge e with a chain of diamonds, say $e \in E(H)$, and let $f : H' \to H$ be the collapse map, $f(X) = \{f \in E(H) : f \in X \vee (f = e \wedge X \cap D_d(e) \neq \emptyset)\}$. For a set of edges, let $C(X) = h_1(X) + l(X)$. We will show that $C(f(X)) \leq C(X)$. First, note that if $e \notin f(X)$, then $X \cong f(X)$ so $C(X) = C(f(X))$. Thus, suppose that $e \in f(X)$, and say that $e = \{u, v\}$. We will reduce to the case that $X \cap D_d(e)$ is a spanning path of $D_d(e)$ by several steps. Once we have done this, then we have a graph which is isomorphic to $\pi(X)$.

First, we remove one of the two sides from each of the diamonds in $X \cap D_d(e)$, as in Figure 0.5 : each such operation decreases h_1 by one, and may increase $l(X)$ by one.[1] If we call X_2 the subgraph of H' obtained by this step, then $C(X_2) \leq C(X)$. Second, we obtain a graph X_3 by adding edges of $D_d(e)$ to X_2 so that $X_3 \cap D_d(e)$ is simple path connecting u and v (and $X_3 \cap H' \setminus D_d(e) = X_2 \cap H' \setminus D_d(e)$). We will now argue that $C(X_3) \leq C(X_2)$, by examining the several cases that are possible: If $e = f(X)$ then, $C(f(X)) = 2 \leq 2$, so we can assume that this is not so. We note that X_2 is connected and has no diamonds by our first step, hence $X_2 \cap D_d(e)$ has one of two homeomorphism types: either an interval, or a disjoint union of two intervals. In the first case, that interval is connected to the opposite side by adding these additional edges. This eliminates at least one leaf, and *may* increase h_1 by one. In the second case, those two intervals cover u and v, thus connecting them with a path decreases the number of leaves by 2 and *may* increase h_1 by one. See Figure 0.6 for an illustration of these cases. Thus, $C(X_3) \leq C(X_2)$, and since $X_3 \cong f(X)$ the claim follows. □

We use the previous lemma to characterize the possible homeomorphism types of $\pi_d(X)$ for $X \in$

[1] Considering the case of a single diamond, we see that it is not true in general that $l(X)$ only increases by at most one if you remove one half of a diamond. However, we started with a lasso, and in particular at each stage of this process X remains connected, and is not just a single diamond, so at most one of the two endpoints of the remaining path of the diamond can become a leaf.

$SCL(D_d(G))$:

Proposition 0.44. *Suppose that Y is a connected graph with $l(Y) + h_1(Y) \leq 2$. Then Y is homeomorphic to one of the following:*

1. *The non-degenerate cases:*
   (a) *A circle S^1.*
   (b) *A lasso.*
2. *The degenerate cases:*
   (a) *A point.*
   (b) *An interval $[0, 1]$.*
   (c) *A lollipop.*
   (d) *A wedge of two circles meeting at a point.*
   (e) *A circle with a chord.*

Proof. We break into cases based on the number of leaves:

1. Suppose that the number of leaves is 0. If $h_1 = 0$, then Y is a (finite) tree with no leaves, which is not possible. If $h_1 = 1$, then Y is a circle. Next, we will show that if $h_1 = 2$ then Y is either a lasso or a circle with a chord, or a wedge of two circles at a point, or a circle with an interval glued on. Observe that from the formula $E - V = h_1 - h_0 = 2 - 1 = 1$, we have that $V = E - 1$. We will show that we smooth[2] down degree 2 vertices to show that Y is homeomorphic to a graph with 1 or 2 vertices. Assume that there at least three vertices in Y. From $h_1 - h_0 = |E| - |V|$ we have that $E = V + 1$. Since there are no leaves, if no vertex has degree 2 then all vertices have degree ≥ 3. This implies that $2E = \sum_v deg(v) \geq 3V$[3], thus $E \geq \frac{3}{2}V$. Using $E = V + 1$, we obtain that $V + 1 \geq (3/2)V$, from which it follows that $2 \geq V$. Thus, if $V \geq 3$, there is a degree 2 vertex. In the case of one vertex (after smoothing), to have $h_1 = 2$, the graph can only be a wedge of 2 circles. We now consider the case of 2 vertices after smoothing, say u and v. First note that there is always an edge between them, and we are left with 2 edges more to distribute. If there is one self loop, say at u, then there must a self loop at v, if we are to have to degree 2 or 1 vertices. If there are no self loops, then we can only run edges between them, and this gives the type of a circle with an interval glued on.
2. Suppose that there is one leaf. If $h_1 = 0$, then Y is a tree with one leaf. This is only possible if Y is a point. If $h_1 = 1$, then we have that Y has exactly as many edges as vertices. We can take an edge e which is part of a leaf node, and delete it. This does not effect the number of cycles. Continuing to do this eventually eliminates the leaf, so we get a cycle. Thus, Y was a lollipop.
3. If there are two leaves, then $h_1(Y) = 0$, hence Y is a tree with two leaves, i.e. a path.

In the next section, we turn to the task of computing the sizes of the set $\pi_d^{-1}(Y)$ for $Y \subseteq E(G)$.

0.6.2. Counting. We let n denote the number of vertices of G. Our goal in this section is to bound $|\pi_d^{-1}(\pi_d(X))|$ as X varies over $SCL(D_d(G))$. The key to these bounds will be the bounds on the number of edges that come from the condition on cycle rank, and the number of vertices. Later on, we will use properties of lollipop ear graphs to further restrict the high probability outcomes.

Lemma 0.45. *Let G be a graph. Let $\pi_d : D_d(G) \to G$ be as in Definition 0.40. Then, we have the following bounds on the sizes of the π_d preimages of connected subgraphs $K \subseteq E(G)$.*

1. *If $h_1(K) \leq 1$, then $|\pi_d^{-1}(K)| \leq (3d^2)^n 2^{dn}$*
2. *If $h_1(K) = 2$ and $l(K) > 0$, then $|\pi_d^{-1}(K)| = 0$.*
3. *If $h_1(K) = 2$ and $l(K) = 0$, and $\pi_d(K)$ does not span G, then $|\pi_d^{-1}(K)| \leq (3d^2)^n 2^{dn}$*
4. *If $h_1(K) = 2$ and K spans G, then $|\pi_d^{-1}(K)| \geq (2^d)^{n+1}$.*

Proof. 1. First, we observe that the condition on h_1 implies that have $|E(K)| \leq |V(K)| \leq n$. The intersection of any SCL with $D_d(e)$ can be either : a lasso , a lollipop, a simple spanning path of $D_d(e)$. This gives an upper bound of $(d^2 2^d + d2^d + 2^d) \leq 3d^2 2^d$ options for the intersection of a SCL with any $D_d(e)$.[4] Thus, $|\pi_d^{-1}(K)| \leq (3d^2)^n 2^{dn}$ follows since the of $D_d(e)$ which a SCL in $\pi_d^{-1}(K)$ intersects in is

[2] That is, if v is a degree 2 vertex, with edges $\{w, v\}$ and $\{v, u\}$ $w \neq u$, then we delete v and add an edge $\{w, u\}$.

[3] Note that a self loop at a vertex v increases its degree by 2.

[4] These are crude upper bounds, derived from the observation that a lasso contained in a $D_d(e)$ consists of two choices of d

$|E(K)| \leq n$

2. This holds because this case is ruled out by Lemma 0.43.
3. We have that $|E(\pi_d(K))| \leq |V(\pi_d(K))| + 1 \leq (n-1) + 1 = n$. This now follows in the same way as the first case.
4. We have at least 2^d options for crossing each edge $D_d(e)$ for $e \in K$, namely the simple paths crossing $D_d(e)$. Since $E - V = h_1 - h_0 = 1$ and K spans G it follows that $|E(K)| = n + 1$. The claim follows.

Lemma 0.46. *If G is a lollipop ear graph, and $K \subset E(G)$ has $h_1(K) = 2$, $l(K) = 0$ and $G[K]$ is connected, and K spans G, then K is a spanning lasso.*

Proof. Since there no leaves, and the graph must span and be connected, all four edges of each lollipop must be in K. To be connected, there must be a path between the two bases of the ears. There can be no edges in the graph other than the path, as otherwise would introduce either leaves or cycles. Thus K is a lasso. Since K is assumed to span, it is a spanning lasso. □

Putting the previous two lemmas together yields:

Lemma 0.47. *Let G be a lollipop ear graph that contains a spanning SCL, J. Set $d \geq (m+2n)^2$, and draw K uniformly from $SCL(D_d(G))$. Then probability that $\pi_d(K)$ is a spanning SCL is at least $\frac{2^m}{1+2^m} \geq 1 - 1/m$.*

Proof. First, by Lemma 0.45 we have that $|\pi_d^{-1}(J)| \geq (2^d)^{n+1}$. From §0.6.2, every $Q \in 2^{E(G)} \setminus SCL(G)$ has either $h_1(Q) \leq 1$ or $h_1(Q) = 2$ but Q does not span G. It follows that that $|\pi_d^{-1}(Q)| \leq (3d^2)^n 2^{dn}$ for every $Q \in 2^{E(G)} \setminus SCL(G)$. Hence, $|\pi_d^{-1}(2^{E(G)} \setminus SCL(G))| \leq (3d^2)^n 2^{dn} 2n^2$. In particular, we have that $\frac{(2^d)^{n+1}}{(3d^2)^n 2^{dn} 2^{n^2}} \geq (2^m)$, by the choice of d and Lemma 0.49. The claim then follows by Lemma 0.48. □

Lemma 0.48. *If $H, N \geq 0$ and $H \geq DN$ for some $D > 0$, then $\frac{H}{H+N} \geq \frac{D}{1+D}$.*

Lemma 0.49. *Fix $q \geq 2$. Then for any $e \in \mathbb{N}$ and $S \geq 1$ if $d \geq 2$ and $d \geq 4(\frac{\log_2(S)+e}{\log_2(q)})^2$ then $q^d \geq Sd^e$.*

Proof. It suffices to pick d so that $\frac{d}{\log(d)} \geq \frac{\log(S)+e}{\log(q)}$. Since $\frac{d}{\log_2(d)} \geq \frac{1}{2}\sqrt{d}$ for $d \geq 2$, the claim follows. □

Proposition 0.50 (Intractability of uniformly sampling simple cycle lassos.). *Suppose that there was a polynomial time Turing machine M and a fixed $\alpha < 1$, such that for each graph G, $M(G)$ output a sample from q_G, where q_G is any probability distribution on $SCL(G)$ with $||q_G - Uniform(SCL(G))||_{TV} \leq \alpha$. Then* RP = NP.

Proof. We show that the assumption leads to an RP algorithm for spanning SCL on lollipop ear graphs, which we have already shown to be NP-hard. Fix $\alpha < 1$, and take $m = \lceil \frac{2}{1-\alpha} \rceil$, and $d = (m+2n)^2$. Algorithm 0.1 supplies the algorithm, and by the assumption on M it runs in polynomial time. Thus, we only have to prove that the algorithm succeeds with the correct error bounds. Since Algorithm 0.1 clearly has no false positives, we only need to check that there is a constant lower bound on the true positive rate. Let $Q(G)$ be the set of spanning SCL of G. We will show that if $|Q(G)| \geq 1$, then the probability of Algorithm 0.1 outputting yes is at least $1/m$. Suppose that q_G is the distribution over 2^E of outputs of Algorithm 0.1 on input M and G, and p_G is the output of lucky guess when *uniformly* sampling from SCL$(D_d(G))$ instead. Suppose that $A = \{C \in SCL(D_d(G)) : \pi_d(C) \in Q(G)\}$. Since $\|p_{D_d(G)} - q_{D_d(G)}||_{TV} < \alpha$, and in particular $p_{D_d(G)}(A) - q_{D_d(G)}(A) < \alpha$, it follow from Lemma 0.47 that $q_{D_d(G)}(A) > p_{D_d(G)}(A) - \alpha \geq 1 - 1/m - \alpha \geq 1/m$. Hence, with probability at least $1/m$, the sample drawn by M from SCL$(D_d(G)$ will land in A. In other words, if $|Q(G)| \geq 1$ and G is a lollipop ear graph, then Algorithm 0.1 will run in polynomial time and answer YES with probability at least $1/m$. Since $Q(G)$ is NP-complete on the class of lollipop ear graphs, it would follow that NP = RP.

places to put the diamond, along with a path which is at most d steps long.

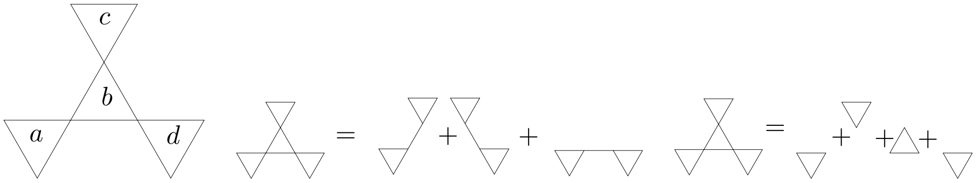


Figure 0.7: Demonstrating non linearity of the support function on the vertices.

Algorithm 0.1 Lucky Guess

```
Construct $D_d(G)$
Let $C$ be an element of SCL($D_d(G)$), chosen with $M$
if $\pi_d(C)$ is a spanning simple cycle of $G$: then
    return YES
else
    Return NO.
```

Theorem 0.51 (Intractability of sampling vertices). *Fix any α, $0 \leq \alpha < 1$. Suppose that there was a polynomial time probabilistic Turing machine M such that for each polytope P, $M(P)$ outputs a sample from q_P, where q_P is any probability distribution on $\mathrm{Vert}(P)$ with $||q_P - Uniform_{\mathrm{Vert}(P)}|| \leq \alpha$. Then $\mathrm{RP} = \mathrm{NP}$. In other words, uniformly sampling vertices of a polytope is intractable.*

Proof. This follows from Proposition 0.50 and subsection 0.4. □

Remark 0.52. *Note that $\mathrm{supp}(v)$is not a linear function on the polytope $P(G)$ (written as in (0.2), not in equational form), even when restricted to the vertices. To see this, consider Figure 0.7, which we think of as a subgraph X of G. We can write the indicator function of the edges of X as 1_X. We can write 1_X as a sum of 3 lassos, L_1, L_2, L_3, or 4 triangles ,T_1, T_2, T_3, T_4. Each of the lassos and the cycles can be written as vertices of P, taking into account that the total weight must sum to 1, and that the bridge edges get twice the weight of the others; in the notation of Definition 0.28 and Figure 0.1 these are the vertices $\frac{1}{8} = f_{L_i}$ and $\frac{1}{3} f_{T_i}$. This leads to $1_X = 4(\sum \frac{1}{8} f_{L_i}) = 3(\sum \frac{1}{3} f_{T_i})$. However, $\mathrm{supp}(f_{L_i}) = 7$ and $\mathrm{supp}(f_{T_i}) = 3$ leads to a contradiction, as $4 * 3 * 7 = 84$, while $3 * 4 * 3 = 36$. Thus, there is no linear function T, such that T restricted to the vertices of $P(G)$ coincides with the support function. We note that this situation on $P(G)$ is unlike the case of $0/1$ polytopes, where supp coincides with the function $\sum x_e$ on the vertices. This is one thing that makes extending the result here to $0/1$ polytopes challenging, as finding a vertex with maximal support on a $0/1$ polytope can be solved by linear programming.*

REFERENCES

[1] S. Aaronson, $\mathrm{P} \overset{?}{=} \mathrm{NP}$, in Open problems in mathematics, Springer, 2016, pp. 1–122.

[2] S. Arora and B. Barak, *Computational complexity: a modern approach*, Cambridge University Press, 2009.

[3] L. N. (https://cstheory.stackexchange.com/users/44995/lorenzo najt), *Can one efficiently uniformly sample a neighbor of a vertex in the graph of a polytope?* Theoretical Computer Science Stack Exchange, https://cstheory.stackexchange.com/q/42705.

[4] R. Impagliazzo and A. Wigderson, P = BPP *unless E has subexponential circuits: derandomizing the XOR lemma*, in Proceedings of the 29th STOC, 1997, pp. 220–229.

[5] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani, *Random generation of combinatorial structures from a uniform distribution*, Theoretical Computer Science, 43 (1986), pp. 169–188, https://doi.org/10.1016/0304-3975(86)90174-X, http://www.sciencedirect.com/science/article/pii/030439758690174X (accessed 2018-11-27).

[6] L. Najt, D. DeFord, and J. Solomon, *Complexity and geometry of sampling connected graph partitions*, arXiv preprint arXiv:1908.08881, (2019).

[7] C. R.Coullard and W. R.Pulleyblank, *On cycle cones and polyhedra*, Linear Algebra and Its Applications, 114/115 (1989), pp. 613–640, https://www.sciencedirect.com/science/article/pii/0024379589904837.

[8] P. Seymour, *Sums of circuits in: Ja bondy, usr murty (eds.), graph theory and related topics*, 1979.

[9] A. J. Sinclair, *Randomised algorithms for counting and generating combinatorial structures*, (1988).